UNIVERSITY OF THESSALY

DOCTORAL THESIS

Design and Experimental Study of Heterogeneous Disaggregated 5G Mobile Wireless Networks

Author:
Nikos MAKRIS

Supervisor:
Assoc. Prof. Thanasis KORAKIS

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

Department of Electrical and Computer Engineering

Dissertation Committee:
Assoc. Prof. Athanasios Korakis
Assoc. Prof. Dimitrios Katsaros
Assist. Prof. Antonios Argyriou

January 2020

Declaration of Authorship

I, Nikos MAKRIS, declare that this thesis titled, "Design and Experimental Study of Heterogeneous Disaggregated 5G Mobile Wireless Networks" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Περίληψη

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Σχεδιασμός και πειραματική μελέτη διαχωρισμένων ετερογενών ασυρμάτων δικτύων 5ης Γενιάς

Νίκος Μακρής

Η πέμπτη γενιά κινητών τηλεπικοινωνιών (5G) συγκεντρώνει πολλές νέες τεχνολογίες τόσο στο δίκτυο πρόσβασης (access network), όσο και στην άκρη του δικτύου (edge) και στο κεντρικό δίκτυο πίσω από την ασύρματη πρόσβαση (Core Network). Η συνδεσιμότητα 5G υπόσχεται σημαντικά υψηλότερη χωρητικότητα του δικτύου (network capacity) με σημαντικά μειωμένη καθυστέρηση μεταδόσεων, με σκοπό να επιτρέπουν σε μια πληθώρα εφαρμογών και κρίσιμων υπηρεσιών να εξυπηρετηθούν πάνω από το νέο αυτό δίκτυο. Με την υιοθέτηση λογισμικού για την υλοποίηση πολλών λειτουργιών του δικτύου (network softwarization), διευκολύνεται η διαχείριση του και η εγκατάσταση μελλοντικών αναβαθμίσεων. Η χρήση λογισμικού για την υλοποίηση υπηρεσιών του δικτύου επεκτείνεται πλέον και στο δίκτυο πρόσβασης χρηστών (Radio Access Network – RAN), με χρήση τεχνολογιών Νέφους (Cloud) για τη δημιουργία Cloud-RANs. Με αυτή την τεχνολογία, κομμάτι του σταθμού βάσης εκτελείται σαν λογισμικό σε υποδομές Νέφους, και μπορεί να υλοποιεί/χειρίζεται διαφορετικές τεχνολογίες για τη πρόσβαση των χρηστών στο δίκτυο. Η υπάρχουσα βιβλιογραφία προτείνει διαφορετικά σημεία για το διαχωρισμό της στοίβας πρωτοκόλλων του σταθμού βάσης, με κάθε σημείο να προσφέρει διαφορετικά πλεονεκτήματα στον πάροχο του δικτύου. Καθώς κινούμαστε πιο χαμηλά στη στοίβα πρωτοκόλλων, σημεία διαχωρισμού χαμηλά στο φυσικό επίπεδο (PHY) επιβάλλουν αυστηρές απαιτήσεις (χαμηλή καθυστέρηση, πολύ υψηλή χωρητικότητα) για τη μετάδοση δεδομένων από το κομμάτι του σταθμού βάσης που εκτελείται στο νέφος προς την κεραία. Αυτό το δίκτυο διασύνδεσης ονομάζεται fronthaul. Ο διαχωρισμός της στοίβας πρωτοκόλλων σε υψηλότερο επίπεδο μπορεί να αποδειχθεί εξαιρετικά ωφέλιμο για τον πάροχο του δικτύου, αφού το δίκτυο fronthaul μπορεί να υλοποιηθεί με χρήση υπαρχουσών τεχνολογιών (πχ. Ethernet) και να επιτρέψει την ενσωμάτωση ετερογενών τεχνολογιών στις τεχνολογίες πρόσβασης. Η ενσωμάτωση υπαρχουσών τεχνολογιών σε διαφορετικά επίπεδα (επίπεδο πρόσβασης χρηστών και στο δίκτυο μεταφοράς) είναι εξαιρετικά σημαντική για την περεταίρω εξέλιξη του δικτύου και τη συμβατότητα με παλαιότερες τεχνολογίες. Σε αυτή την εργασία, σχεδιάσαμε και μελετήσαμε πρωτοποριακούς τρόπους λειτουργίας διαχωρισμένων σταθμών βάσης, με στόχο την ενσωμάτωση ετερογενών τεχνολογιών στο επίπεδο πρόσβασης χρήστη. Οι ερωτήσεις που προσπαθούμε να απαντήσουμε είναι οι ακόλουθες: 1) Ποιο είναι το καλύτερο σημείο για το διαχωρισμού της στοίβας πρωτοκόλλων ενός σταθμού βάσης; 2) Πως μπορούμε να ενσωματώσουμε ετερογενείς τεχνολογίες στον διαχωρισμένο σταθμό βάσης; 3) Πως μπορούμε να εξυπηρετήσουμε χρήστες με πολύ χαμηλή καθυστέρηση μετάδοσης πακέτων; 4) Αφού αυτές οι αρχιτεκτονικές βασίζονται αρκετά σε λογισμικό, πως μπορούμε να ενορχηστρώσουμε τη λειτουργία τους με αποδοτικό τρόπο;

Αρχικά, ξεκινάμε με την υπάρχουσα αρχιτεκτονική για κινητές επικοινωνίες και σχεδιάζουμε και μελετάμε τον διαχωρισμό της στοίβας πρωτοκόλλων σε διαφορετικά σημεία. Με βάση το σημείο διαχωρισμού, διαφορετικές απαιτήσεις υπάρχουν για το δίκτυο μεταφοράς από το κομμάτι που υλοποιεί τα χαμηλότερα επίπεδα μέχρι το νέφος. Μιας και στόχος μας είναι η ενσωμάτωση υπαρχουσών τεχνολογιών για το δίκτυο μεταφοράς (Ethernet/WiFi), σχεδιάζουμε και αποτιμούμε πειραματικά διαχωρισμούς για την στοίβα πρωτοκόλλων που μπορούν να χρησιμοποιήσουν εμπορικό εξοπλισμό. Το δίκτυο μεταφοράς (fronthaul) βασίζεται σε πακέτα (packetized) και δεν εξαρτάται από ακριβό υλικό με νεότερα πρωτόκολλα (πχ. το Common Public Radio Interface – CPRI). Με τη χρήση του ανοιχτού λογισμικού OpenAirInterface (OAI), υλοποιούμε και αποτιμούμε πειραματικά διαφορετικά σημεία διαχωρισμού σε πραγματικό περιβάλλον.

Στη συνέχεια, ασχολούμαστε με το υψηλότερο επίπεδο διαχωρισμού της στοίβας (στο υψηλότερο επίπεδο 2) και μελετάμε την ενσωμάτωση διαφορετικών τεχνολογίών για το δίκτυο πρόσβασης χρηστών. Ο διαχωρισμός σε αυτό το επίπεδο αναφέρεται στα πρότυπα για το πρωτόκολλο πρόσβασης δικτύου 5G New Radio (NR), και είναι ανάμεσα στο επίπεδα Packet Data Convergence Protocol (PDCP) και Radio Link Control (RLC). Αυτός ο διαχωρισμός δημιουργεί δυο διαφορετικές μονάδες, την κεντρικοποιημένη (Centralized Unit – CU) που εκτελείται στο νέφος, και την κατανεμημένη (Distributed Unit – DU) που εκτελεί τα πρωτόκολλα στο χαμηλότερο επίπεδο 2 και επίπεδο 1 της στοίβας πρωτοκόλλων. Εντούτοις, το επίπεδο PDCP έχει χρησιμοποιηθεί και στο παρελθόν για την ενσωμάτωση ετερογενών τεχνολογιών σε σταθμούς βάσης, όπως για παράδειγμα με το πρωτόκολλο LTE-WiFi Aggregation Adaptation Protocol (LWAAP) για την ενσωμάτωση WiFi πρόσβασης. Αυτό δημιουργεί τις συνθήκες για την ενσωμάτωση ετερογενών τεχνολογιών στο δίκτυο ως κατανεμημένες μονάδες (DUs). Επεκτείνουμε την αρχική μελέτη διαχωρισμού σταθμών βάσης, ώστε να ενσωματώσουμε τεχνολογίες WiFi. Με την εισαγωγή ενός διαχειριστή φόρτου δικτύου, μπορούμε να επιλέξουμε για κάθε μετάδοση πακέτου ποια κατανεμημένη μονάδα (και επομένως διαφορετικής τεχνολογίας) μπορεί να χρησιμοποιηθεί για κάθε πελάτη του δικτύου. Επεκτείνουμε την αρχική πρωτότυπη υλοποίηση στην πλατφόρμα OAI και πειραματιζόμαστε σε πραγματικές συνθήκες. Στη συνέχεια παρουσιάζουμε αποτελέσματα για την απόδοση του δικτύου για διαφορετικές παραμέτρους (πχ. απόσταση CU/DU, το πρωτόκολλο επιπέδου μεταφοράς που χρησιμοποιείται, κλπ.)

Ωστόσο, τέτοιες υποδομές συχνά προσφέρουν υπερ-πυκνή ανάπτυξη του δικτύου, με πολλαπλές ετερογενείς τεχνολογίες να χρησιμοποιούν το ίδιο ασύρματο φάσμα. Επομένως, πρέπει να υπάρξει αποδοτικός συντονισμός για τη διασφάλιση της σωστής λειτουργίας όλων των τεχνολογιών στο επίπεδο πρόσβασης. Γι' αυτόν το σκοπό, σχεδιάζουμε και μελετάμε νέα σηματοδοσία για τον συντονισμό ετερογενών τεχνολογιών σε τοπολογίες διαχωρισμένων σταθμών βάσης. Επεκτείνουμε την υλοποίηση μας με ένα νέο μηχανισμό επικοινωνίας που μεταφέρει παραμέτρους της διαμόρφωσης του δικτύου και στατιστικά χρήσης, για την διεξαγωγή συμπερασμάτων για τη χρήση του φάσματος σε μια περιοχή. Με βάση τις συλλεχθείσες μετρήσεις, εφαρμόζουμε ένα αλγόριθμο για τον καθορισμό της χρήσης φάσματος από κάθε τεχνολογία στην ίδια μπάντα συχνοτήτων. Αποτιμάμε τη νέα λύση μέσω πειραματισμού σε πραγματικό περιβάλλον και παρουσιάζουμε τα αποτελέσματα μας που αποδεικνύουν καλύτερη λειτουργία του υπό μελέτη δικτύου.

Σαν επόμενο βήμα, μελετάμε και σχεδιάζουμε νέα λειτουργικότητα σε αρχιτεκτονικές διαχωρισμένων σταθμών βάσης με ετερογενείς τεχνολογίες, με σκοπό την επίτευξη χαμηλής καθυστέρησης μετάδοσης για την πρόσβαση υπηρεσιών από τους χρήστες του δικτύου. Η τεχνολογία Multi-access Edge Computing (MEC) έχει προταθεί σαν μέθοδος για τη δραστική μείωση της καθυστέρησης πρόσβασης σε υπηρεσίες που παρέχονται πάνω από το δίκτυο, με την τοποθέτηση των υπηρεσιών αυτών κοντά στην άκρη του δικτύου. Η υπάρχουσα βιβλιογραφία προτείνει διαφορετικές τοποθεσίες για την τοποθέτηση των υπηρεσιών αυτών, με αυτές να τοποθετούνται στην καλύτερη περίπτωση μαζί με την κεντρικοποιημένη μονάδα (CU) και να χειρίζονται κίνηση που κατευθύνεται προς το κεντρικό δίκτυο (backhaul δίκτυο). Με βάση τον διαχωρισμένο σταθμό βάσης, μελετάμε και σχεδιάζουμε την τοποθέτηση υπηρεσιών στο δίκτυο fronthaul, είτε μαζί ή κοντά στις κατανεμημένες μονάδες του δικτύου (DUs). Τα πειραματικά μας αποτελέσματα δείχνουν πως αναφορικά με την καθυστέρηση, ακόμα και παλαιότερες τεχνολογίες (όπως η τεχνολογία LTE της 4ης γενιάς δικτύων) μπορεί να εξυπηρετήσουν αρκετές από τις εφαρμογές που έχουν αναπτυχθεί για δίκτυα 5ης γενιάς.

Τέλος, αφού το δίκτυο μας εξαρτάται σε μεγάλο βαθμό από λογισμικό, μελετάμε και σχεδιάζουμε την ενορχήστρωση του σαν εικονικές συναρτήσεις δικτύου (Virtual Network Functions - VNFs). Η αρχιτεκτονική Network Functions Virtualization Management and Orchestration (NFV-MANO) παρέχει μια προτυποποιημένη μέθοδο για τη διαχείριση και την εύκολη ανάπτυξη (εικονικών) υπηρεσιών. Η αρχιτεκτονική NFV-MANO αρχικά επικεντρώνεται στην ανάπτυξη υπηρεσιών/εφαρμογών πάνω από datacenters. Όμως, η εισαγωγή αρχιτεκτονικών δικτύου που εξαρτώνται σε μεγάλο βαθμό από λογισμικό, ακόμα και για το ασύρματο δίκτυο, παρέχουν έδαφος για την αλλαγή του τρόπου με τον οποίο διαχειριζόμαστε το υλικό. Με αυτή την έννοια, σχεδιάζουμε την επέκταση της οντότητας Virtual Infrastructure Manager (VIM) της αρχιτεκτονικής ώστε να χειρίζεται εικονοποιημένα ασύρματα δίκτυα, που δημιουργούμε σε πραγματικό περιβάλλον. Ο σχεδιασμός μας επιτρέπει την εισαγωγή των επεκτάσεων αυτών με διαφάνεια στο σύστημα, ώστε να επιτρέψουν τη μεταφορά υπηρεσιών από άλλες πλατφόρμες που δεν υποστηρίζουν ασύρματες τεχνολογίες. Εφαρμόζουμε τις επεκτάσεις μας αυτές σε πραγματικό περιβάλλον, και αποτιμάμε την πλατφόρμα μας σχετικά με την απόδοση της, χρησιμοποιώντας σαν υπηρεσίες δικτύου τους διαχωρισμένους σταθμούς βάσης με πολλαπλές τεχνολογίες ασύρματης πρόσβασης που προηγουμένως αναπτύξαμε.

UNIVERSITY OF THESSALY

Abstract

Department of Electrical and Computer Engineering

Design and Experimental Study of Heterogeneous Disaggregated 5G Mobile Wireless Networks

by Nikos MAKRIS

The fifth generation of mobile networking (5G) is fostering several advancements in the access, edge and core network, promising to offer significantly higher network capacity with lower latency over the network, allowing a variety of applications and critical services to thrive around this ecosystem. Extensive softwarization of the network adds up to the flexibility of management and eases future upgrades. Softwarization expands even to the Radio Access Network (RAN) through the emerging concept of Cloud-RAN. In such setups, part of the base station stack is running as a software unit that can be instantiated in the Cloud, controlling different radio access modules for providing network access to the network. Relevant literature suggests different points for splitting the base station stack, with each point bringing different benefits for the operator. As we move down to the network stack, splits in the lower physical layer pose stringent requirements (very low latency, very high capacity) for the transport of signaling and data to the cloud-located unit (fronthaul network). Splits taking place at higher layers may prove to be more beneficial for the operator, as they can be served with existing network infrastructure (e.g. Ethernet links), and also allow the integration of heterogeneous technologies to the cell. Integration with existing technologies at different levels (access and transport network) is crucial for the continuous evolution of the platform and compatibility with legacy protocols. In this thesis, we design, study and introduce novel functionality in disaggregated base station setups, towards enabling the operation of heterogeneous technologies in the user access. The fundamental questions that we try to answer are the following: 1) What is the best point to disaggregate the base station stack? 2) How can heterogeneous technologies be integrated in the network cell? 3) Given such a disaggregated infrastructure, how can we serve end users with low latency? 4) Since such infrastructure is highly softwarized, how can we efficiently orchestrate its operation?

Initially we begin with the cellular stack, and study and design its disaggregation at different points. Depending on the point that the stack is dissaggregated, different requirements are posed on the transport network, from the radio unit to the cloud. As we also target integration with existing technologies for the transport network (e.g. Ethernet/WiFi), we design and develop splits for the stack that can be

accommodated using off-the-shelf equipment. We use a packetized transport network (base station fronthaul), in order not to rely on expensive hardware or newer protocols (e.g. the Common Public Radio Interface - CPRI). We develop and implement our splits for the stack in the open source framework OpenAirInterface (OAI), and experimentally evaluate different splits in a testbed setup.

Following this, we delve into the higher OSI layer 2 split for the cellular stack and design and study the integration of different technologies to the cell. The higher layer splits that are currently standardized for the 5G New Radio (NR) protocol, assume the disaggregation of the stack at the same point, between the Packet Data Convergence Protocol (PDCP) and the Radio Link Control (RLC) layers. This disaggregation creates two different units, the Centralized Unit (CU) running at the Cloud, and the Distributed Unit (DU) running the lower layer 2 and Layer 1 protocols. Nevertheless, the PDCP layer has been used in the past for integrating different technologies in the cell, such as for example the LTE-WiFi Aggregation Adaptation Protocol (LWAAP). This creates the circumstances for the integration of heterogeneous technologies in the cell as DUs. Based on our prior study, we design and integrate WiFi as a radio access technology to the base station cell. Using a load-balancing controller, we can manage in a per packet basis which DU can serve each network client. We evaluate our contributions by extending the OAI prototype implementation and experiment in real settings with the prototype. Following this, we provide performance metrics when tuning different parameters (e.g. distance between CU/DU, transport protocol used, etc.).

Nevertheless, such infrastructures tend to offer ultra-dense network deployment, with multiple technologies being collocated within the same spectrum. Therefore, efficient coordination needs to take place in order to ensure the proper operation of all the accommodated technologies. To this aim, we design and study a spectrum coordination scheme that can be applied in such disaggregated heterogeneous base station architectures. We design signaling that carries radio configuration parameters and statistics, in order to conclude on the spectrum usage of a specific area. Based on the collected measurements, we apply an algorithm for efficiently placing all the technologies within the same frequency band. We evaluate our design by further extending the implementation and experimenting in a real testbed environment with densely deployed wireless networks. Our results illustrate higher performance of the under-study network.

As our next step, we design and study new features in such multi-technology disaggregated networks, towards providing low-latency service access for users connected to the cell. Multi-access Edge Computing (MEC) has been proposed as a means to drastically minimize the service access latency for clients connected to the network, by placing the services closer to the network edge. Existing literature suggests different placements for the MEC services, which at the best case is collocated with the CU of the network, intercepting backhaul traffic. Given the disaggregated nature of the cell, we design and study how the placement of services on the fronthaul network, collocated with/close to the DUs of the network, can reduce the service access latency. Our testbed experiments denote that in terms of latency, even legacy technologies (e.g. 4G LTE) can be suitable for several of the proposed 5G applications.

Finally, as this network is highly softwarized, we design and study the orchestration of the software as Virtual Network Functions (VNFs). Network Functions Virtualization Management and Orchestration (NFV-MANO) provides a standardized approach on the management and effortless deployment of (virtual) services. Although NFV-MANO initially focused on the deployment of services over datacenters, the introduction of fully softwarized network architectures even for the wireless part creates fertile ground for the re-conception of the manner through which the underlying hardware is managed. In this concept, we design extensions to a Virtual Infrastructure Manager (VIM) in order to handle virtualized wireless network interfaces, hosted on the generic networking nodes of the testbed. We design the extensions in a manner that allows their transparent introduction to the existing operation of the platform, thus allowing portability of network services and network functions to other instances as well. We implement and apply our design in a testbed environment, and subsequently benchmark the framework in terms of performance and evaluate it by using the disaggregated multi-technology base stations that we have previously developed.

List of Publications

The results, ideas and figures of this thesis have been included in the following publications:

Book Chapters

[B1] A. Tzanakaki, M. Anastasopoulos, N. Gomes, P. Assimakopoulos, J. M. Fabrega, M. Svaluto Moreolo, L. Nadal, J. Gutierrez, V. Sark, E. Grass, D. Camps-Mur, A. De la Oliva, N. Molner, X. Costa Perez, J. Mangues, A. Yaver, P. Flegkas, N. Makris, T. Korakis, D. Simeonidou. **Transport Network Architecture**, *5G System Design: Architectural and Functional Considerations and Long Term Research*, 2018, 151-180, John Wiley & Sons.

[B2] N. Makris, T. Korakis, V. Maglogiannis, D. Naudts, N. Nikaein, G. Lyberopoulos, E. Theodoropoulou, I. Seskar, C. A. Garcia Perez, P. Merino Gomez, M. Tosic, N. Milosevic, S. Spirou. **FLEX: A Platform for 4G/5G Wireless Networking Research, Targeting the Experimentally-Driven Research Approach**, *Building the Future Internet through FIRE*, 2017, 111-154, River Publishers.

Journals and Magazines

[J1] N. Makris, V. Passas, T. Korakis. **Pairing MEC and CloudRAN: placing services on the Fronthaul of Heterogeneous Networks**, *submitted to IEEE Communications Magazine*.

[J2] N. Makris, C. Zarafetas, A. Valantasis, T. Korakis. **Service Orchestration over Wireless Network Slices: Testbed Setup and Integration**, *submitted to IEEE Transactions on Network and Service Management*.

[J3] V. Nejkovic, F. Jelekovic, N. Makris, V. Passas, T. Korakis, M. Tosic. **Semantic Coordination On the Edge of Heterogeneous Ultra Dense Networks**, *submitted to Springer Computing Journal*.

Conferences

[C1] N. Makris, V. Passas, C. Nanis, T. Korakis. **On Minimizing Service Access Latency: Employing MEC on the Fronthaul of Heterogeneous 5G Architectures**, in *IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), 2019*, IEEE.

[C2] N. Makris, P. Karamichailidis, C. Zarafetas, T. Korakis. **Spectrum Coordination for Disaggregated Ultra Dense Heterogeneous 5G Networks**, in *European Conference on Networks and Communications (EuCNC), 2019*, IEEE.

[C3] D. Camps-Mur, K. Katsalis, I. Freire, J. Gutierrez, N. Makris, S. Pontarelli, R. Schmidt. **5G-PICTURE: A Programmable Multi-Tenant 5G Compute-RAN-Transport Infrastructure**, in *European Conference on Networks and Communications (EuCNC), 2019*, IEEE.

[C4] N. Makris, V. Passas, T. Korakis, L. Tassiulas. **Employing MEC in the Cloud-RAN: An Experimental Analysis**, in *Technologies for the Wireless Edge Workshop of ACM Mobicom), 2018*, ACM.

[C5] N. Makris, C. Zarafetas, A. Valantasis, T. Korakis, L. Tassiulas. **Integrating NFV-MANO with Wireless Services: Experiences and Testbed Development**, in *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2018*, IEEE.

[C6] N. Makris, C. Zarafetas, P. Basaras, T. Korakis, N. Nikaein, and L. Tassiulas. **Cloud-based Convergence of Heterogeneous RANs in 5G Disaggregated Architectures**, in *IEEE International Conference on Communications (ICC), 2018*, IEEE.

[C7] N. Makris, A. D. Samaras, V. Passas, T. Korakis, L. Tassiulas. **Measuring LTE and WiFi coexistence in Unlicensed spectrum**, in *European Conference on Networks and Communications (EuCNC), 2017*, IEEE.

[C8] N. Makris, P. Basaras, T. Korakis, N. Nikaein, L. Tassiulas. **Experimental Evaluation of Functional Splits for 5G Cloud-RANs**, in *IEEE International Conference on Communications (ICC), 2017*, IEEE.

Demonstrations

[D1] V. Passas, N. Makris, C. Nanis, T. Korakis. **MEC service placement over the Fronthaul of 5G Cloud-RANs**, in *IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), 2019*, IEEE.

[D2] N. Makris, C. Zarafetas, K. Choumas, P. Flegkas, T. Korakis. **Virtualized Heterogeneous 5G Cloud-RAN deployment over Redundant Wireless Links**, in *IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), 2019*, IEEE.

In addition, our research efforts within the same period led to the following publications that are not directly related to this thesis:

Conferences

[C1] V. Passas, N. Makris, V. Miliotis, T. Korakis. **Pricing Based MEC Resource Allocation for 5G Heterogeneous Network Access**, in *IEEE Global Communications Conferences (Globecom), 2019*, IEEE.

[C2] K. Chounos, N. Makris, T. Korakis. **Enabling Distributed Spectral Awareness for Disaggregated 5G Ultra-Dense HetNets**, in IEEE 5G World Forum, 2019, IEEE.

[C3] V. Passas, V. Miliotis, N. Makris, T. Korakis. **Dynamic RAT Selection and Pricing for Efficient Traffic Allocation in 5G HetNets**, in *IEEE International Conference on Communications (ICC), 2019*, IEEE.

[C4] V. Passas, N. Makris, V. Miliotis, T. Korakis, L. Tassiulas. **MATCH: Multiple Access for multiple Traffic Classes in 5G HetNets**, in *IEEE International Conference on Communications (ICC), 2018*, IEEE.

[C5] C. Zarafetas, N. Makris, A. Apostolaras, T. Korakis, L. Tassiulas. **Flexible Cross-Technology Offloading Using SDN**, in *13th IEEE TELSIKS, 2017*, IEEE.

[C6] V. Passas, V. Miliotis, N. Makris, T. Korakis, L. Tassiulas. **Paris Metro Pricing for 5G HetNets**, in *IEEE Global Communications Conferences (Globecom), 2016*, IEEE.

Acknowledgements

As this long endeavor has reached its end, I would like to express my deep gratitude to a group of people who provided my with guidance, assistance, support, and encouragement. It is certain that without their help, I would not have been able to complete my PhD studies.

First and foremost, I would like to deeply thank my supervisor, Assoc. Prof. Thanasis Korakis, as well as Prof. Leandros Tassiulas, who have provided me with guidance, motivation and support throughout the years of my studies in University of Thessaly, and gave me the opportunity to work in such an inspiring research team. Thanks to their continuous support and my exposure to different research ideas and projects, I was able to broaden my knowledge in different aspects. I am also grateful to the members of my dissertation committee, Assoc. Prof. Dimitrios Katsaros, Assist. Prof. Antonios Argyriou, Assoc. Prof. Gerasimos Potamianos, Assoc. Prof. Spyridon Lalis, Prof. Symeon Papavassiliou and Assist. Prof. Anna Tzanakaki for accepting to serve in the examination committe of my thesis. To Assist. Prof. Navid Nikaein I want to express my sincere thanks for guiding me through the early implementations of the thesis during the first year.

I would also like to thank all my friends and lab-mates in NITLab for their support and collaboration everyday. Especially I would like to thank Christos Zarafetas with whom I have been working and collaborating since the very beginning in the lab. Also the rest of the guys, Virgilios Passas, Alexandros Valantasis, Stratos Keranidis, Vasilis Miliotis, Giannis Kazdaridis, Polychronis Symeonidis, Ilias Syrigos, Kostas Chounos, Apostolis Apostolaras, Dimitris Giatsios, Kostas Choumas, Harris Niavis, Donatos Stavropoulos, Antonis Kalkanof, Ellie Efstathiou, Giannis Mavridis, Panagiotis Karamichailidis and Christos Nanis.

My sincere thanks to my closest friends, Achilleas, Antonis, Giannis, Christos, Spyros, Kostas and Leonidas for their encouragement all these years. They have been at my side for more than twenty years now, always encouraging and supporting each other in every aspect.

Last but not least, I want to thank my family, my daughter Eleni, my wife Natasa, my parents Eleni and Fotis and my brother Kostas. I have no words to express my gratitude for them, their support, patience and unconditional love for all my decisions. This dissertation is dedicated to them.

Dedicated to my family.

Contents

List of Figures

List of Tables

List of Abbreviations

3GPP	**3**rd **G**eneration **P**artnership **P**roject
AP	**A**ccess **P**oint
API	**A**pplication **P**rogramming **I**nterface
APN	**A**ccess **P**oint **N**ame
AR	**A**ugmented **R**eality
BBU	**B**ase**B**and **U**nit
BH	**B**ack**H**aul
BS	**B**ase **S**tation
C-RAN	**C**loud **RAN**
CAPEX	**Cap**ital **Ex**penditure
CN	**C**ore **N**etwork
CPRI	**C**ommon **P**ublic **R**adio **I**nterface
CU	**C**entralized **U**nit
DL	**D**own**L**ink
DU	**D**istributed **U**nit
E-UTRAN	**E**volved **UMTS** **T**errestrial **R**adio **A**ccess **N**etwork
eCPRI	**e**volved **C**ommon **P**ublic **R**adio **I**nterface
EEA	**E**PS **E**ncryption **A**lgorithm
eICIC	**e**nhanced **I**nter-**c**ell **I**nterference **C**oordination
eNB	**e**volved **N**ode**B**
EPC	**E**volved **P**acket **C**ore
EPS	**E**volved **P**acket **S**ystem
ESSID	**E**xtended **S**ervice **S**et **Id**entifier
F1AP	**F1** **A**pplication **P**rotocol
FEC	**F**orward **E**rror **C**orrection
FFT	**F**ast **F**ourier **T**ransform
FH	**F**ront**H**aul
FIRE	**F**uture **I**nternet **R**esearch and **E**xperimentation
gNB	**g**igabit **N**ode**B**
GTP	**G**PRS **T**unneling **P**rotocol
GW	**G**ate**w**ay
HSS	**H**ome **S**ubscriber **S**ervice
HTB	**H**ierarchical **T**oken **B**uffer
ICIC	**I**nter-**c**ell **I**nterference **C**oordination
IP	**I**nternet **P**rotocol

LAN	**L**ocal **A**rea **N**etwork
LBO	**L**ocal **B**reak**O**ut
LTE	**L**ong **T**erm **E**volution
LWA	**L**TE **W**LAN **A**ggregation
LWAAP	**L**TE **W**LAN **A**ggregation **A**daptation **P**rotocol
LXC	**L**inu**X** **C**ontainer
MAC	**M**edium **A**ccess **C**ontrol
MANO	**M**nagement **an**d **O**rchestration
MCS	**M**odulation and **C**oding **S**cheme
MEC	**M**ulti-access **E**dge **C**omputing
MME	**M**obility **M**anagement **E**ntity
mmWave	**m**illi**m**eter **Wave**
MPD	**M**edia **P**resentation **D**escription
MTU	**M**aximum **T**ransferable **U**nit
NF	**N**etwork **F**unction
NFV	**N**etwork **F**unctions **V**irtualization
NGFI	**N**ext **G**eneration **F**ronthaul **I**nterface
NGMN	**N**ext **G**eneration **M**obile **N**etwork
NITOS	**N**etwork **I**mplementation **T**estbed using **O**pen **S**ource platforms
NR	**N**ew **R**adio
NSD	**N**etwork **S**ervice **D**escriptor
OAI	**O**pen**A**ir**I**nterface
OMF	c**O**ntrol and **M**anagement **F**ramework
OPEX	**Op**erational **Ex**penditure
OSI	**O**pen **S**ystems **I**nterconnection
OSM	**O**pen **S**ource **M**ANO
OvS	**O**pen **vS**witch
PDCP	**P**acket **D**ata **C**onvergence **P**rotocol
PDN-GW	**P**acket **D**ata **N**etwork **G**ate**w**ay
PDN	**P**acket **D**ata **N**etwork
PDU	**P**rotocol **D**ata **U**nit
PHY	**Phy**sical layer
QoE	**Q**uality **o**f **E**xperience
QoS	**Q**uality **o**f **S**ervice
RAN	**R**adio **A**ccess **Network**
RAT	**R**adio **A**ccess **T**echnology
RCC	**R**adio **C**loud **C**ontroller
RLC	**R**adio **L**ink **C**ontrol
RNTI	**R**adio **N**etwork **T**emporary **I**dentifier
RRC	**R**adio **R**esource **C**ontrol
RRH	**R**emote **R**adio **H**ead
RRU	**R**emote **R**adio **U**nit

RTT	**R**ound **T**rip **T**ime
S1AP	**S**1 **A**pplication **P**rotocol
SAP	**S**ervice **A**ccess **P**oint
SCTP	**S**tream **C**ontrol **T**ransmission **P**rotocol
SDAP	**S**ervice **D**ata **A**daptation **P**rotocol
SDN	**S**oftware **D**efined **N**etworking
SDR	**S**oftware **D**efined **R**adio
SDU	**S**ervice **D**ata **U**nit
SGW	**S**erving **G**ate**w**ay
SSID	**S**ervice **S**et **Id**entifier
TBS	**T**ransport **B**lock **S**ize
TC	**T**raffic **C**ontrol
TCP	**T**ransmission **C**ontrol **P**rotocol
TEID	**T**unnel **E**ndpoint **ID**dentifier
UDP	**U**nreliable **D**atagram **P**rotocol
UE	**U**ser **E**quipment
UHD	**U**SRP **H**ardware **D**river
UL	**U**plink**L**ink
UMTS	**U**niversal **M**obile **T**elecommunications **S**ystem
USRP	**U**niversal **S**oftware **R**adio **P**eripheral
VAP	**V**irtual **A**ccess **P**oint
VDU	**V**irtual **D**ata **U**nit
VIM	**V**irtual **I**nfrastructure **M**anager
VLAN	**V**irtual **LAN**
VM	**V**irtual **M**achine
VNF	**V**irtual **N**etwork **F**unction
VNFD	**VNF D**escriptor
VR	**V**irtual **R**eality
WiFi	**Wi**reless **Fi**delity
WLAN	**W**ireless **LAN**

Chapter 1

Introduction

Contents

1.1 Motivation and Problem Statement

Cloud-based Radio Access Network (Cloud-RAN or C-RAN) processing has been identified as one of the major enablers for 5G mobile network access. It relies on the disaggregation of the base station stack to two or more units and running part of it as a service in the Cloud. It is the natural evolution of distributed Base Station (BS) units, allowing higher flexibility and ease of new feature deployment, as most of the processing is done through software in the Cloud. Initially, when the concepts of Cloud-RAN were introduced, the base station baseband processing was expected to be moved to the cloud as a service running Layers 1, 2 and 3 in the Baseband Unit (BBU). The access part of the network was expected to be implemented through a Remote Radio Head (RRH), considered to be a rather passive element of the network. This concept was reflected as the full centralization of the base station units, as shown in Figure 1.1. A partial centralization is considered when the RRH includes the baseband processing (e.g. the entire L1), and is annotated as a Remote Radio Unit (RRU), and thus considered an active element of the network.

This shift towards disaggregated base station setups has proven to bring multple benefits to network operators, in terms of CAPEX and OPEX cost reduction, as well as the network operation and support of advanced functions. Some of these are presented below, according to [114]:

- *Creating an energy-efficient/green networking infrastructure*: Through centralized processing of the Cloud-RAN architecture, the number of base station sites can be reduced. Thus the site support equipment can be drastically reduced, inducing energy savings. Moreover, the distance from the RRHs to the UEs can be decreased since the cooperative radio technology can reduce the interference among RRHs and allow a higher density of RRHs. Smaller cells with lower transmission power can be deployed while the network coverage quality is not affected. The energy used for signal transmission will be reduced, which is especially helpful for the reduction of power consumption in the RAN and extend the UE battery time. Lastly, a pool of BBUs can be considered as a shared resource among a large number of virtual BS, athus allowing a much higher utilization rate of processing resources to be achieved.

- *Cost-saving on CAPEX & OPEX*: Since BBUs are aggregated, it is much easier for centralized management and operation, saving a lot of the Operations and Management associated costs. Secondly, although the number of RRHs may not be reduced in a C-RAN architecture their functionality is much simpler, thus saving development costs. Moreover, the size and power consumption of the RRHs are both reduced and they can be mounted on poles with minimum site support and management.
- *Enhanced Capacity for the provided network*: In C-RAN, virtual base stations may work together in a large physical BBU pool and they can easily share the signaling, traffic data and channel state information (CSI) of active UEs in the system. It is much easier to implement joint processing and scheduling to mitigate inter-cell interference (ICI) and improve spectral efficiency.
- *Network Adaptability to Non-uniform Traffic*: C-RAN is also suitable for non-uniformly distributed traffic due to the load-balancing capability in the distributed BBU pool. As the potential coverage of a BBU pool is larger than the traditional monolithic base stations, non-uniformly distributed traffic generated from UEs can be distributed in a virtual base station unit that sits in the same BBU pool.
- *Smart Traffic Offloading*: By enabling the smart breakout technology in C-RAN, the growing internet traffic from portable devices can be offloaded from the core network of operators. The benefits include reduced backhaul traffic and cost, reduced core network traffic and gateway upgrade cost, reduced latency to the users and differentiation of traffic among users.

Nevertheless, depending on the point and the functionalities that are integrated in the RRU, the demands in capacity and latency for the transport network (RRU to BBU path) change. Low layer splits, where the baseband processing is located entirely in the Cloud require high capacity and low latency links for the fronthaul network - the network between the BBU and the RRU. Thus different protocols have been proposed, as a means to ensure reliable transfer of time-sensitive data. The most outstanding paradigm is the Common Public Radio Interface (CPRI), proposed

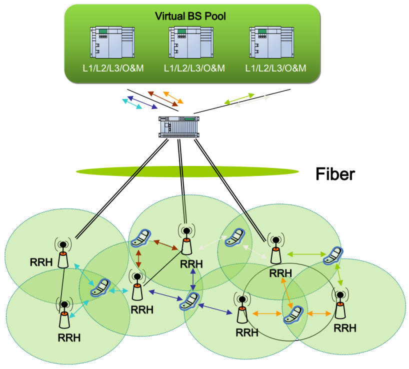


FIGURE 1.1: Fully centralized CloudRAN architecture

for meeting the demands of a highly time-sensitive fronthaul interface. Nevertheless, the adoption of such solutions for the fronthaul requires costly upgrades and deployments of fiber equipment, to cope with the increased demands for low latency and high capacity. Thus, this fact deprives the first and foremost advantages of C-RAN technology, the reduced CAPEX and OPEX cost of the platform.

To this aim, the Next Generation Fronthaul Interface (NGFI) consortium was formed to identify and model the fronthaul network for different splits of the stack. In 2015, China Mobile research team released their whitepaper [114] where they identify different splits for the cellular stack which can be accommodated in a packetized fronthaul, able to be transported over existing technologies (e.g. Ethernet). Figure 1.2 shows the 6 initially proposed splits that can be accomodated over such a fronthaul interface, whereas tables 1.1 and 1.2 the calculated fronthaul capacity and compression ratio for the data for fronthauling an 8-antenna TD-LTE base station. The required latency for all the splits is below 1 msec, apart from the Interface 1 which has more slack requirements and can operate with latency times up to 100 msec.

TABLE 1.1: Fronthaul interface requirements per each NGFI proposed split for Interfaces 1-3

	Interface 1		Interface 2		Interface 3	
	Bandwidth	Ratio	Bandwidth	Ratio	Bandwidth	Ratio
Downlink	174 Mbps	1	179.2 Mbps	1	125.2 Mbps	1
Uplink	99 Mbps	1	78.6Mbps	1	464.6 Mbps	6

TABLE 1.2: Fronthaul interface requirements per each NGFI proposed split for Interfaces 4-5

	Interface 4		Interface 5	
	Bandwidth	Ratio	Bandwidth	Ratio
Downlink	498 Mbps	3	9830.4 Mbps	66
Uplink	2689.2 Mbps	36	9830.4	131

As this work was based only on modeling data, one of the goals of this thesis was to study and experimentally evaluate different splits for the cellular base station stack.

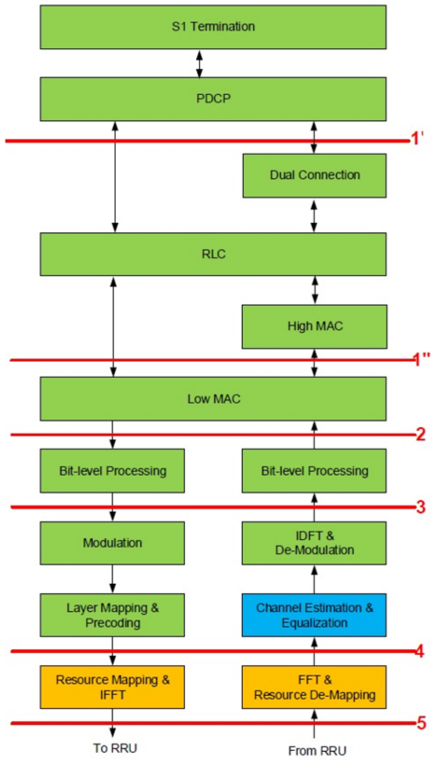


FIGURE 1.2: NGFI split options for the cellular base station stack

Other goals are to design and study how heterogeneous technologies can be integrated into the cell, and how service access latency can be minimized to host 5G applications. Finally, as the 5G architecture is highly softwarized, we targetted in the design of efficient orchestration mechanisms for virtual base stations. From the implementation perspective of the platform, as at the time of the thesis no 5G New Radio (NR) implementation was available, we used the Long Term Evolution (LTE) protocol. As the protocol characteristics for the higher layers are almost identical, and we measure the performance of the fronthaul interface, our results are expected to show minimal deviations when applied to NR. Below we present some of our experimental tools that we use throughout this thesis, and following these, we summarize the contribution of each chapter.

1.2 Experimental Tools and Methods

In this section, we describe the main tools that we use for evaluating the experimental solutions that we introduce. These are broken down in the following: 1) NITOS, the wireless testbed that we use, located in University of Thessaly, Greece, 2) the OpenAirInterface platform, used to form real cellular base stations over commodity hardware, and 3) the Open Source MANO platform, used for the orchestration of software services as Virtual Network Functions (VNFs). Below we detail each component.

(A) Indoor testbed nodes

(B) Outdoor testbed

(C) Macroscale base station serving mobile users of the testbed

FIGURE 1.3: NITOS testbed and equipment

1.2.1 NITOS Testbed

The NITLAB team at the University of Thessaly is operating since 2007 the Network Implementation Testbed using Open Source platforms (NITOS), which has evolved over the years to a compact solution for evaluating bleeding-edge ideas on the forefront of networking-related research. The NITOS testbed is one of the largest single-site open experimental facilities in Europe, allowing users from around the globe to take advantage of highly programmable equipment, which is remotely accessible. The testbed is an integral part of larger federations of resources, such as OneLab [31] and Fed4FIRE [105], enabling experiments with more heterogeneous resources. NITOS has an established user base of over 4000 users in the past years, with over 20 researchers using the infrastructure daily. In short, the current offering of the testbed is the following:

- Over 100 nodes equipped with IEEE 802.11 a/b/g/e/n/ac compatible equipment, and using open-source drivers. The nodes are compatible also with the IEEE 802.11s [42] protocol for the creation of wireless mesh networks. The nodes feature multiple wireless interfaces, and are high-end computers, with quad-core Intel Core i5 and Core i7 processing capabilities, 4/8 GBs of RAM and SSD disks.

- Commercial off-the-shelf (COTS) LTE testbed, consisting of a highly programmable

LTE macrocell, multiple femtocells, an experimenter configurable Evolved Packet Core (EPC) network and multiple User Equipment (UE), such as USB dongles and Android Smartphones [67].

- Open Source LTE equipment, running over commodity Software Defined Radio (SDR) equipment, by the adoption of the OpenAirInterface platform [83] (`www.openairinterface.org`). The platform is allowing multiple configurations for creating highly customizable beyond 4G networks.
- COTS WiMAX testbed, based on a highly programmable WiMAX base station in standalone mode (no ASN-GW component), along with several open-source WiMAX clients.
- A Software Defined Radio (SDR) 5G testbed, consisting of 10 USRPs N210, 12 USRPs B210, 4 USRPs X310 and 4 ExMIMO2 FPGA boards. MAC and PHY algorithms can be executed over the SDR platforms, with very high accuracy.
- A millimeter-wave testbed, operating in the V-band (60GHz), based on six nodes [104]. The platforms support high data-rate point-to-point setups, with beam steering capabilities of up to 90 degrees with a step of 7.5 degrees.
- The nodes are interconnected with each other via 5 OpenFlow [75] hardware switches, sliced using the FlowVisor [98] framework.
- A Cloud Computing testbed, consisting of 96 Cores, 286 GB RAM and 10 TBs of hardware storage. For the provisioning of the cloud, OpenStack [97] is used.
- Multiple WSN clusters, supporting the IEEE 802.15.4, 802.11 and LoRaWAN protocols, gathering measurements such as temperature, luminosity, air quality, radiation emission, etc.

The equipment is distributed across three different testbed locations in the city of Volos, and can be combined for creating a very rich experimentation environment. The nodes are running any major UNIX based distributions.

1.2.2 The OpenAirInterface Platform

OpenAirInterface (OAI) wireless technology platform as the first open-source software-based implementation of the LTE system spanning the full protocol stack of 3GPP standards. It features contributions both in E-UTRAN (wireless access) and the Evolved Packet Core (EPC) [83]. It can be used to build and customize an LTE base station and core network on a PC and connect a commercial UEs to test different configurations and network setups and monitor the network and mobile device in realtime. OAI is based on a PC hosted software radio frontend architecture. With OAI, the transceiver functionality is realized via a software radio front end connected to a host computer for processing. This approach is similar to other software-defined radio (SDR) prototyping platforms in the wireless networking research community such as SORA [103]. OpenAirInterface is the first fully x86-based SDR solution in open-source, providing both UE, eNB, and core-network functionality. OAI is written in standard C for several real-time Linux variants optimized for x86 and released as free software under the terms of version 3 of the GNU General Public License (GPLv3). OAI provides a rich development environment with a range of built-in tools such as highly realistic emulation modes, soft monitoring and debugging tools, protocol analyzer, performance profiler, and configurable logging system for all layers and channels.

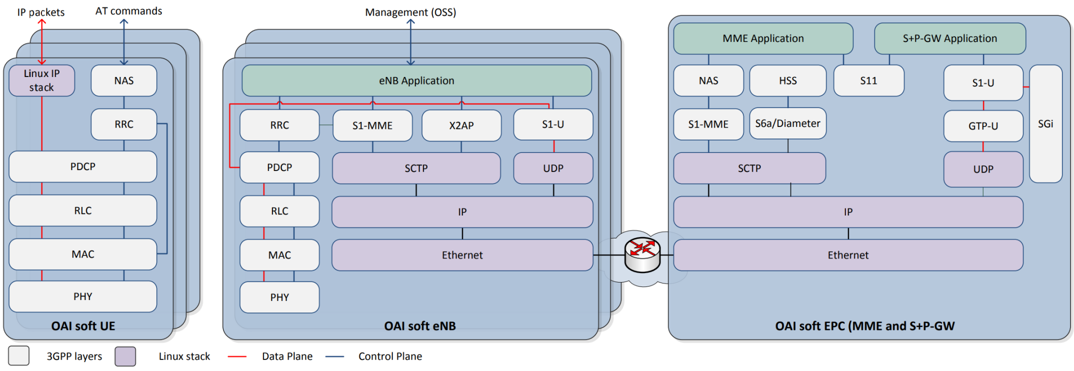


FIGURE 1.4: OpenAirInterface protocol stack (Figure appearing in [48].

In this thesis, we make extended use of the OAI platform at different levels. All of our contributions are introduced as features in the OAI platform, that enable the integration of new data plane signaling for the communication of disaggregated architectures, as well as integration with non-3GPP technologies (WiFi) to the cellular stack. All the solutions are wrapped as VNFs and are deployed in the testbed, allowing the real-world execution of experiments.

1.2.3 Open Source MANO

Towards enabling experimentation with emerging 5G tools, we adopted the Open Source Mano orchestrator for providing experimenters with a fully-fledged solution for network virtualization and NFV functions. Open Source MANO (OSM) [30] is an open source Management and Orchestration (MANO) software stack closely aligned with ETSI NFV reference architecture that meets the requirements of commercial NFV networks. Its primary focus is to accelerate the implementation of network virtualization. OSM at its core enhances interoperability over several components (e.g. VNFs, VIMs, and SDN controllers) and facilitates a plugin platform that creates a scalable MANO environment. It combines cutting edge technology such as Containers [100] and Juju [47], to facilitate the required plugin framework. Specifically, OSM is composed of 3 fundamental components:

- The Service Orchestrator (SO) integrates the RIFT.io orchestration engine and is responsible for end-to-end service orchestration and provisioning. It stores the specifications for the network services, i.e., the VNF definitions and NS catalogs, and can manage and monitor the status of already deployed services.
- The Resource Orchestrator (RO) is used to provide services over a particular Infrastructure as a Service (IaaS) provider, e.g., the NITOS testbed. In OSM release FIVE, RO can deploy network services over various platforms, namely Amazon Web Services, OpenStack, VMware, Whitestack's WhiteCloud and OpenVIM. The SO and RO components can be jointly mapped to the NFVO entity in the ETSI MANO architecture [28].
- The VNF Configuration and Abstraction (VCA) module performs the initial VNF configuration via the use of Juju Charms.

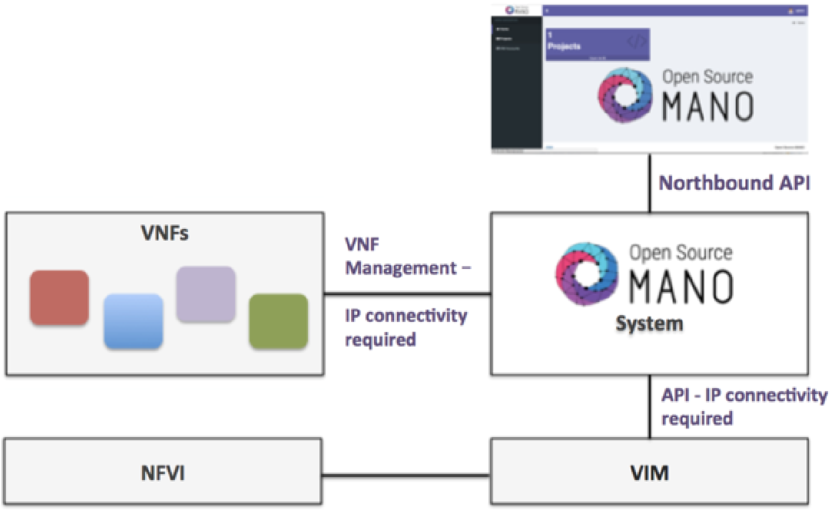


FIGURE 1.5: Open Source MANO topology (figure courtesy of [30])

In order to deploy a software service through OSM, a Network Service Descriptor (NSD) is needed. The NSD specifies the network function components and the relations between them to be deployed over the IaaS provider. OSM defines these specifications for the NSD and VNF Descriptors (VNFD) via YAML-based documents. Specifically, the SO handles the descriptors to define a VNF in terms of deployment and operational requirements. Virtual Deployment Units (VDUs) are represented as VMs or Containers and connections between them are specified in the VNFD via the internal Virtual Links (VLs). OSM uses archived format both for NSD and VNFD. This archive consists of the service/VNF description, initial configuration scripts, and other auxiliary details. More details on the process and our contributions to the platform are provided in Chapter 6.

1.3 Thesis Synopsis

Based on the aforementioned architecture for C-RAN, and the existing experimentation and software tools, this thesis provides real-world experimentally-driven results for such disaggregated 5G architectures. The fundamental questions that we try to answer are the following: 1) What is the best point to disaggregate the base station stack? 2) How can heterogeneous technologies be integrated into the network cell? 3) Given such a disaggregated infrastructure, how can we serve end users with low latency? 4) Since such infrastructure is highly softwarized, how can we efficiently orchestrate its operation? In the following paragraphs, we provide a summary of the works included in this thesis, towards addressing these questions.

Initially, in **Chapter 2** we begin by studying and experimentally evaluating real Cloud-RAN deployments, with respect to different functional splits. We use as a reference architecture the 3GPP LTE stack and argue about the functional split applicability in contemporary networks. We study and design OSI layer 2 functional splits, that can be used for the convergence of multiple heterogeneous wireless technologies in an all-in-one unit. We focus on splits taking place at the higher layer 2, or between layer 2 and layer 1. By deploying our approach in a real testbed setup, we extract the fronthaul network transfer requirements for the different splits and present our experimental findings, compared with the respective simulation results.

Following this, in **Chapter 3** we focus on the higher layer 2 splits, between the Packet Data Convergence Protocol (PDCP) and Radio Link Control (RLC) layers. This disaggregated RAN split defines a Centralized Unit (CU) and multiple Distributed Units (DUs) is considered As this split is expected to bring numerous advantages to mobile network operators, through the isolation of the stack from the PDCP layer and upwards, the CU is considered to be able to act as the Cloud-based convergence point among multiple heterogeneous technologies in the provisioned networks and hence able to serve multiple heterogeneous DUs. Moreover, data rate requirements for this type of split are not very demanding, thus allowing the IP-based transferring of data from the DU to CU and vice-versa. In this chapter, we propose, design, implement and evaluate a protocol for a Cloud-RAN based architecture allowing the selection and dynamic switching of different heterogeneous networks in the RAN. We rely on the open-source OpenAirInterface platform and extend it to support data plane splitting of the LTE functionality, and the subsequent data injection to WiFi networks. We evaluate the platform using a real network setup, under several scenarios of network selection and different delay settings.

In **Chapter 4** we study the case of ultra-dense deployments of disaggregated base station setups. Network densification and integration of heterogeneous technologies enable larger network capacity through the aggregation of multiple links, thus assisting the transition from the existing network infrastructure to innovative 5G networks. Nevertheless, as Ultra-Dense Heterogeneous Networks may operate in the same wireless spectrum, their performance potential may be hindered through the operation in overlapping frequencies. Thus, efficient coordination is required between the involved heterogeneous technologies. In this chapter, we consider a disaggregated base station setup, with capabilities to incorporate heterogeneous technologies for serving the UEs. We design and develop signaling between the heterogeneous Distributed Units and the Central Unit and apply a spectrum coordination algorithm for optimal use of the wireless spectrum.

Chapter 5 presents our contributions towards minimizing service access latency through Multi-access Edge Computing. Multi-access Edge Computing (MEC) has been proposed as the means to drastically minimize the service access latency, by bringing computational resources and services closer to the wireless network edge. Edge resources are planned for extended usage in the upcoming 5G networks, as they can meet stiff latency demands required from services being developed around this ecosystem (e.g. VR, e-Health, Industry 4.0, etc.). At the same time, 5G networks redefine the operation of traditional base station units, by disaggregating them and operating part of them in the Cloud, thus creating Cloud-RANs. These Cloud-RANs can also be heterogeneous, based on our prior contributions, allowing users to access the network through multiple wireless technologies. In this chapter, we blend the novel disaggregated and heterogeneous base station architecture with the MEC concept and propose the deployment of the edge computing services even closer to the network edge. We study and design the cases when services can be placed close or over the machines hosting the radio access services for the network access. By exploiting features for integrating heterogeneous radio resources in the cell, we create a switching technology for the MEC side of the network that selects the technology through which each client of the network is served. We integrate our cotnributions in a real-world prototype, and present our findings. Our extensive experiments highlight the efficiency of our scheme in terms of latency, as well as for Quality of Experience when streaming dynamically adaptive video.

Finally, in **Chapter 6**, we study the orchestration of our software developed services for the disaggregated heterogeneous base stations. We adopt the Network Functions Virtualization Management and Orchestration (NFV-MANO) architecture, which provides a standardized approach for the management and effortless deployment of (virtual) services. Although NFV-MANO initially focused on the deployment of services over datacenters, the introduction of fully softwarized network architectures even for the wireless part creates fertile ground for the re-conception of the manner through which the underlying hardware is managed. In this chapter, we consider the case of the NITOS wireless testbed and adopt the Open Source MANO framework for provisioning virtual services on top of the equipment. We design and introduce extensions the VIM service for the testbed to handle virtualized wireless network interfaces, hosted on the generic networking nodes of the testbed. The extensions are introduced transparently to the existing operation of the platform, to allow the portability of network services and network functions to other instances as well. We focus on providing virtual functions which are internetworked over wireless links, which traditionally are not handled by the framework, and allow easier interaction of the end-users with the testbed. We benchmark the framework in terms of performance and evaluate it with a use case around the single-click deployment of heterogeneous disaggregated base stations.

Chapter 2

Development and evaluation of functional splits for C-RANs

Contents

2.1 Chapter Introduction

The concept of centralizing the Radio Access Networks (RANs) has been identified as one of the keen aspects to be adopted by the 5th Generation (5G) mobile networks. RAN centralization lies in the simple concept of decoupling the baseband processing from the actual radio interface, and providing it as a Cloud Computing service. Such Cloud services can enable many real-time operations that are currently not viable in the "full-stack" base stations, and can serve as the main enablers for the virtualization of the wireless network, allowing multi-tenancy over the actual same hardware resources.

Cloud-RAN lies on separating the Base Band Units (BBUs) from the actual RF front-end. Existing base station deployments are using a more decentralized approach, where the base station units process a very large part of the networking stack. This fact can pose a big obstacle in the base station syncing, especially for processes like the enhanced Inter-Cell Interference Control (eICIC) for LTE-Advanced. Yet, they can all be dealt with when the BBU processes for different RANs are closely located in the Cloud.

Furthermore, the disaggregation of the functions from a single unit to the BBU and a Remote Radio Unit (RRU) can fit extremely the 5G concepts of network virtualization and higher delivered capacity per user in a single geographical area [109]. The RRU can be considered as either a passive element, with the sole purpose to transmit low-level data over the air (Remote Radio Head - RRH) or a more intelligent unit, where part of the processing takes place (e.g. the entire PHY layer or parts of it). Based on the demand and an existing pool of BBUs in the cloud, the serving RRUs covering a single area can be instantiated on the fly. Moreover, since network slicing and virtualization can be handled beyond the RRU, as its sole purpose is only to transmit low-level raw data (ideally raw IQ samples for the RRH case), multiple operators can take advantage of the very same physical equipment. Network virtualization can take place as a virtual function in the Cloud, thus enabling multiple tenants (mobile network operators) to take advantage of the same network equipment.

For the implementation of Cloud-RAN architectures, high bandwidth connections are needed from the RRU to the BBU. Depending on the point where the split takes place, the transfer requirements of the network may vary; this highly depends on the back/front- hauling technology as well. Employing an IP based scheme can induce delays for the processing and packetization of data, as well as the maximum number of served RRU units [19]. Deduced from these requirements, very stringent delay times need to be met for the realization of the splits. Moreover, interoperability of the Cloud-RAN with the existing network infrastructure is highly desired as well. This essentially means that existing fiber-based infrastructure and protocols (e.g. CPRI) or copper links can be exploited for the realization of Cloud-RANs. Copper links have come to the fore due to their high availability, as well as flexibility of protocols that can be executed over a packetized data plane. This is also reflected in the standardization activities of IEEE P1914.3 for enabling Radio over Ethernet communication, mainly for the Cloud-RAN applications.

Towards addressing this emerging research challenge, the Next Generation Fronthaul Interface (NGFI) has identified in [114] the possible splits for the BBU/RRU functions. Both high- and low-level splits have been identified, with the LTE architecture as a reference.

This chapter's main contributions are the following:

- To extract the real-time transfer requirements for a 5G Cloud-RAN in the FH.
- To implement and evaluate different functional splits over the LTE networking stack, complying with NGFI.
- To experimentally evaluate different transport protocols for the aforementioned splits (UDP/TCP/SCTP).

The splits that we evaluate take place at two different points of Layer 2 of LTE stack; 1) PDCP/RLC and 2) MAC/PHY. We employ the open-source platform OpenAirInterface [82] for the realization of the splits and evaluate our solutions in a real environment, when using a 1Gbps Ethernet link for our FH.

The rest of the chapter is organized as follows. Section 2.2 is providing an overview of any previous related work and our motivation. Section 2.3 is discussing our choice for the functional splits, as well as the pros and cons of each solution. Section 2.4 is presenting our contributions and experimental setup, whereas in Section 2.5 we showcase our experimental findings. Finally, Section 2.6 concludes our work and presents some future directions.

2.2 Motivation and Related Work

Cloud-RAN has been identified as one of the key 5G enablers. Next Generation Mobile Networks (NGMN) alliance has pinpointed the advantages, as well as potential interfaces for facilitating the functional splits in [7]. The advantages of employing a centralized processing unit, located in the cloud has been also described in [109]. The authors argue on the Cloud-RAN applicability for 5G schemes, as well as analyze the transfer requirements for the fronthaul (FH), when the functional splits take place at different points of the PHY layer of LTE. Moreover, in [20], the different technologies that are available for realizing the Cloud-RAN architectures are illustrated. Potential splits are identified along with the technologies employed for the data transportation to the Cloud.

Similarly, authors in [9] detail the requirements for the FH network, concerning low-level splits. An analysis of the potential technologies used for FH and BH of 5G networks, based on these specific requirements for PHY layer functional splits is presented in [10].

A study resembling our contributions is presented in [25]. The authors identify high-layer and low-layer splits for FH and extract the transfer requirements for the network. Yet, the work relies on simulation-based models for the network setup, while in all of the aforementioned cases the authors do not consider the existing legacy networks as potential technologies for transferring the data to the Cloud.

In order to use existing packet-based networks, instead of circuit-based fiber connections (e.g. CPRI), the extra delays of packet encapsulation, decapsulation, and processing have to be taken into consideration. Authors in [19] and [79] analyze and model these requirements using IP based networks for PHY layer splits.

Yet, experimentally driven results are very scarce regarding the Cloud-RAN modeling. Authors in [80] present a platform where the RRU is composed of all the LTE PHY layer functions, whereas the rest of the eNB processing is taking place as a separate process executed in the Cloud, based on the OpenAirInterface platform.

Following up this work, authors in [8] present through real experiments the delay that is incurred when the BBU is operating inside different virtualization environments (e.g. KVM, LXC, etc.). Similarly, authors in [16] present their platform for Cloud-RANs.

In this work, we present our contributions to the same Open Source platform for LTE, in which we implement functional splits at two different layers. Our work differentiates from similar former studies in the fact that it is, to the best of our knowledge, the first experimentally driven work on characterizing this type of splits over an IP based FH network. By employing both simulation and real testbed experiments, we evaluate and extract the real-time requirements for the operation of such a Cloud-RAN architecture. We use an IP-based BH, and measure the limitations induced by the packetization and processing of different protocols used for transferring the data. We use two approaches for the data transport: 1) based on stateless protocols (UDP), for the PHY layer split, as data are more delay-sensitive regarding the scheduling of the transmissions and 2) stateful protocols (TCP/SCTP) for higher layer splits, as they can operate with more slack delay requirements, if proper buffering of the data is employed.

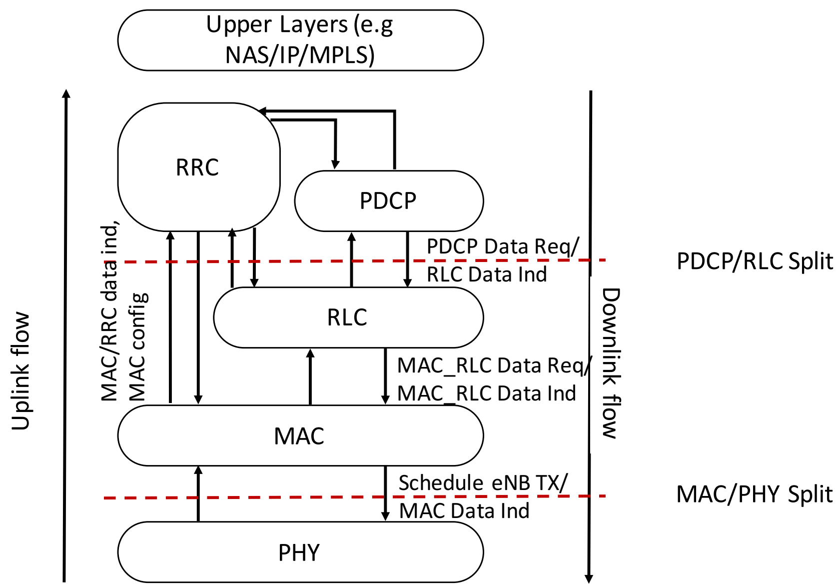


FIGURE 2.1: LTE Reference Architecture and Identified Functional Splits

2.3 Functional Split Architecture

We employ the LTE protocol stack as our reference architecture in order to identify our functional splits and conduct our experiments. In this section, we provide a brief overview of the functions of each layer in a bottom-up manner, and the potential of each split when deployed in real systems.

PHY layer is dedicated to the transmission and reception of control and user data over the air. This may include functions such as FEC, encoding/decoding, equalization, FFT and finally the D/A or A/D conversion. Functional splits can be identified at different points of the PHY layer, used mainly for fronthauling the LTE network. MAC layer is endowed with the scheduling processes and allocating resources for the served UEs in the network. Once a stream is scheduled for transmission in a specific subframe in the MAC layer, it is delivered to PHY. RLC is a sublayer used to transfer the higher layer PDUs to MAC SDUs, by concatenating/segmenting them and reassembling them. PDCP is used as the interface with the IP based networks, used to do packet compression and removing the IP header before giving the packets to lower layers for scheduling their transmission over the air.

Yet, although the splits other than the ones dedicated inside the PHY layer may yield only small performance benefits for Cloud-RAN applications [109], they can be the enablers for novel applications for 5G. The splits that are dealt with in this work are the following:

- **PDCP/RLC split:** Splits over the MAC layer seem to be yielding only small performance benefits for 5G, as they could presumably need more transmissions over the FH link in order to send the same amount of data to the RRU. The data sent are actual IP packets after the PDCP processing, which have not gone through the concatenation process of the RLC. A qualitative disadvantage of such layer splits is that the data sent to the RRU might need significantly more transmissions over the network, as it is of lower size than the ones outputted by the RLC process. Yet, as most of the contemporary networks can transfer packets of up to a specific size (e.g. MTU equal to 1500 bytes), the usage of technologies like Ethernet can be advantageous for the functional splits. Although this might seem a drawback for this type of split, there are clear benefits of using the PDCP as a convergence layer among different technologies [92]; multiple technologies can be coordinated from a single PDCP/IP instance at the base station, enabling seamless mobility experience across several technologies, with very little overhead for the network operator.
- **MAC/PHY split:** The MAC/PHY split that we examine has been identified as one of the potential splits in [109], [19] and [9]. In this case, the RRU and BBU are synced and operate on a subframe basis. The BBU unit can instruct, based on the output of the MAC scheduling policy, the subframe allocation for each UE. The actual data that needs to be transferred from the BBU to the RRU is equal to the Transport Block Size (TBS), depending on the modulation and the physical resource blocks which are allocated to each specific UE. This split can be beneficial for the real-world application of several algorithms and technologies, such as dynamic scheduling of multiple RRUs, spectrum coordination algorithms [11], beamforming coordination [22], etc.

Figure 2.1 is illustrating the architecture and the splits that we evaluate. We employ the OpenAirInterface platform as our reference implementation of the LTE stack in order to choose the functions which will be split. Regarding the PDCP/RLC split, the splitting function takes place in the following manner: whenever PDCP is receiving a packet, it goes through its normal procedure before being relayed to the next layer. As soon as the packet is processed, it is sent to the RRU where RLC processes it. The resulting stream is placed in a buffer waiting for the MAC protocol to send a request for it. It is worth to mention here that the existing buffer for handling this type of data in OpenAirInterface needed to be extended for carrying out our experiments.

Regarding the MAC/PHY split, we choose to override the part where the two layers communicate with each other; this is the point where upon the end of the MAC scheduling algorithm, the BBU instructs the RRU at which subframe the data will be transmitted over the air. This means that no buffering of the packets takes place inside the RRU, but are solely handled by the eNB application (BBU). Whenever data needs to be sent over the air, data streams are sent to the RRU along with all the signaling needed to orchestrate the PHY layer, including the subframe scheduled for transmitting, the number of physical resource blocks, the modulation and coding scheme (MCS), the antennas, etc.

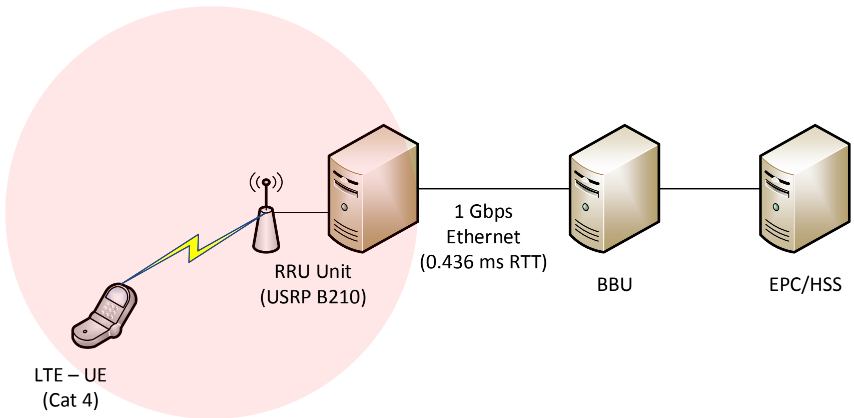
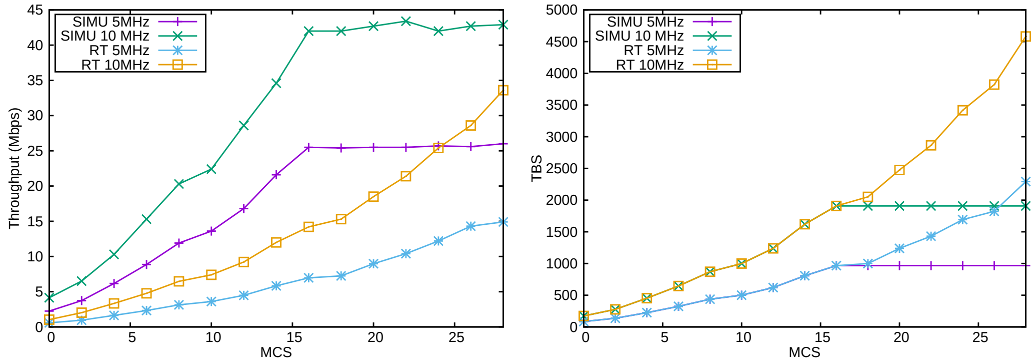


FIGURE 2.2: Experiment setup for the split evaluation

2.4 Experimental Setup

For the evaluation of the chosen splits, we experiment using the NITOS environment. NITOS is a heterogeneous testbed located at the premises of University of Thessaly, in Greece. It offers a very rich experimentation environment with resources spanning from commercial LTE, to WiFi and Software Defined Radio platforms [67]. The topology that we employ is depicted in Figure 2.2. We split the eNodeB process of OpenAirInterface into two parts, one being executed on a node with a USRP B210



(A) Throughput achieved per MCS profile

(B) TBS used per MCS profile

FIGURE 2.3: Reference results taken with OpenAirInterface for simulation (SIMU) and real time (RT) operation

platform, being our RRU, and one on a dedicated testbed node. For the two under-study functional splits, different parts of the code are either executed over the BBU or the RRU. A third testbed node is running the EPC and HSS software, while we also employ a fourth node equipped with an LTE Cat. 4 UE.

The splits are configured as follows; we override the default processes for OpenAirInterface and configure one listening server and one client for two different binaries of the code. Each time that a packet is about to be sent as a data request from a higher to a lower layer, it gets packed in a standard message and is sent over the link. We use the *socat* application for redirecting the traffic over the network to the listening server. In case of a data indication message (lower to higher layer direction), a similar process takes place. The parameters of the LTE network setup, as well as the different scenarios that we use, are shown in Table 2.1. For all of our cases, the testbed nodes are static, and the UE is reporting values of excellent signal quality, with the RSRP ranging from -76 to -83 dBm and reported RSSI values up to -53 dBm. We perform our experiments using two different bandwidth settings, for LTE channels of 5 and 10 MHz.

TABLE 2.1: Testbed and Simulation parameters

Network Parameters	**Values**
LTE mode	FDD Band 7
LTE Frequency	2680 MHz (DL)
No RBs	25, 50
UE	Cat. 4 LTE, Huawei E3272
OAISIM channel emulation	Rayleigh
OAISIM mobility	STATIC
Backhaul/Fronthaul RTT	$\sim$ 436 msec
Backhaul/Fronthaul connection	1Gbps Ethernet
Ethernet MTU size	1400 bytes

Similarly with the real network setup, we use the same testbed nodes in order to run the OpenAirInterface emulation platform [107] (OAISIM). All the functional splits are implemented for both setups, real-time and emulator. Regarding OAISIM, we use the setup where the PHY layer is abstracted, meaning that certain functions of the PHY are omitted. This setup is able to yield better results, as the wireless channel is modeled using predefined patterns. For all of our simulation experiments, the multipath model used is Rayleigh, as it is the one that is used by default in OAISIM. The splits are taking place over the same network as happens with the real setup.

As our FH network is an IP based one, we choose to evaluate the performance of different protocols for the splits, depending on the split and real-time requirements of the network. Although stateless protocols are the ones that should be adopted for this type of experiments (UDP), we also incorporate TCP and SCTP as our transport solution for the FH for the cases of PDCP/RLC split, which has more loose delay requirements. Our experiments demonstrate that fronthauling is viable also with these solutions, although more capacity for the FH network is needed for achieving similar performance as with the UDP solutions. Regarding the TCP experiments, the congestion control algorithm that we use is *Cubic*, as the rest of the algorithms yielded worst performance results, indicated also in [108]. For the SCTP results, we use 5 parallel streams for each association and do not use the multi-homing features. We provide experimental results with a resolution of 10 for each measurement. For generating traffic for our measurements, we use the *iperf* traffic generator, set to saturate the wireless link with UDP traffic.

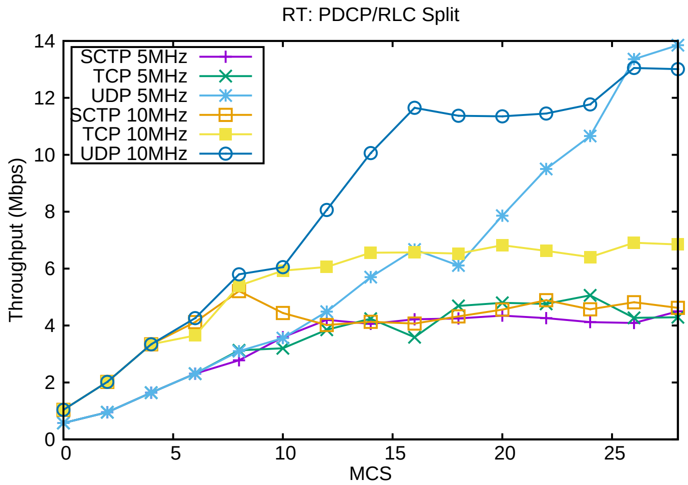
In the following section, we present our experimental results, obtained by running the aforementioned functional splits in a real testbed environment as well as with simulation results. The evaluation is broken down into three subsections. Initially, we briefly provide some reference measurements from the under study platform without implementing any split. Following, we showcase the experimental results for the PDCP/RLC split, and finally we present our results regarding the MAC/PHY split. Although the splits are applicable for both the Downlink and Uplink data flow, we present measurements for the Downlink channel, as it is the one with the most stringent requirements for transfer. We measure and comment on the total achieved throughput for the LTE UE, for the two under examination functional splits.

2.5 System Evaluation

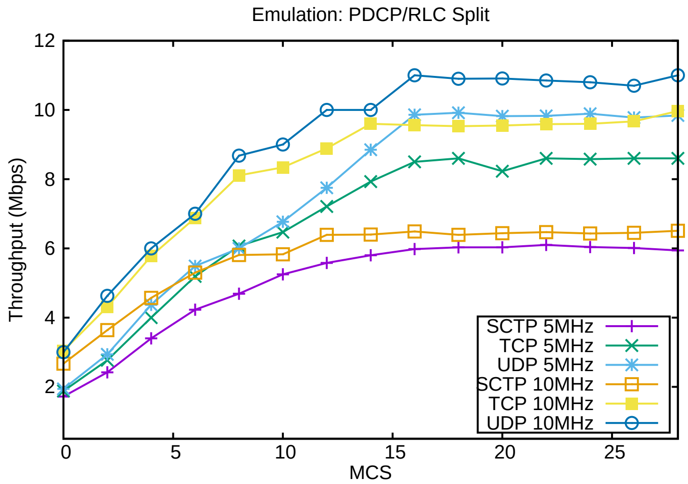
2.5.1 Reference Measurements

Initially, we present some benchmarking results of the platform that we use for our experiments. In this setup, we use the *vanilla* OpenAirInterface platform, configured as either the LTE emulation platform (OAISIM) or set to operate in real-time (RT), running the whole LTE stack in one base station binary application.

In Figure 2.3 we provide the results regarding the throughput performance achieved per each MCS profile allocated by the eNodeB scheduler (Fig. 2.3a), as well as the



(A) Split results for real time



(B) Split results for OAISIM

FIGURE 2.4: PDCP/RLC splits when using UDP, TCP and SCTP for fronthauling

mean TBS used (Fig. 2.3b). TBS is of paramount importance for the MAC/PHY split, as the output of the MAC processing mandates the transferring of equal-sized data to the RRU within the time scheduled for transmission. The bits that are allocated by the LTE scheduler for transmission are the ones that will define the bottleneck in our under investigation fronthaul network.

We observe that the OAISIM platform yields the same results for MCS indexes over 16. This happens due to the abstraction flags that are passed to the emulation platform, which omits the execution of certain PHY-layer blocks in favor of better performance. Similarly, the TBS allocated for each transmission follows the same pattern.

2.5.2 Evaluation of PDCP/RLC splits

Since the PDCP functions happen at a higher layer, the real-time operation can be maintained if proper buffering is used at the RLC level. PDCP is processing every incoming IP packet and upon the header compression, it delivers it via the FH network to the RRU implementing the LTE protocol below RLC. Whenever MAC layer is finished with the scheduling of its buffered packets, it requests the RLC buffered packets. Based on this fact, real-time operation can not be broken if other than stateless protocols are used for the FH. Nevertheless, this fact means that larger memory allocation is needed for enabling such a split. For our experiments, we extended the memory allocation for both the BBU and RRU applications, in order to reassure that the machine does not run out of memory.

Figure 2.4 is presenting our experimental results when using the real-time platform. As we can observe for the real-time operation (Figure 2.4a), and concentrating on the 5MHz transmissions, we see that the worst-performing protocol is SCTP. Although SCTP has been introduced as a protocol resolving the head-of-line blocking effect that is present in TCP, its implementations for the Linux kernel are not that mature compared with TCP. For 5MHz, the bottleneck for SCTP when fronthauling the LTE data over the 1Gbps Ethernet link is around 4Mbps. The same bottleneck exists for SCTP fronthauling for channels with 10MHz bandwidth.

Regarding TCP experiments, we see that the bottleneck for transferring the 5MHz channels is happening around MCS 18, meaning for TBS sizes over 1000 bits. Similarly, for the 10MHz transmissions, the bottleneck is around MCS 14. Regarding the UDP experiments, for both 5 and 10 MHz transmissions, the bottleneck for the 1Gbps FH link is around 13Mbps. UDP outperforms both SCTP and TCP as it due to its stateless nature, the overhead that is posed on the fronthaul only regards the transmission of IP packets, after the PDCP handling and compression to the remote RRU with the RLC layer.

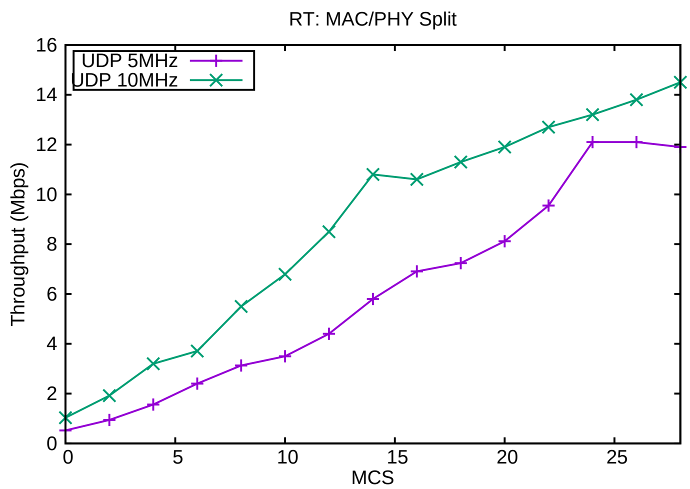


FIGURE 2.5: MAC/PHY real time splits

Regarding the simulation results, a similar pattern as in the real time experiments is witnessed. We observe that the throughput achieved by OAISIM is bounded at approximately 11Mbps for the best case, when using UDP for transferring the data.

As illustrated in Figure 2.3a, for MCS indexes over 16, the data is sent using the same TBS, as several of the PHY functions are omitted in favor of better performance.

2.5.3 Evaluation of MAC/PHY splits

Following the PDCP/RLC splits, we conduct experiments regarding the lower layer split. We present results only for the UDP based data flow, as our first set of experiments denoted that it is the protocol that achieves better performance in such splits. Moreover, the RRU employs a minimal queueing mechanism, so whenever the data is sent over the fronthaul to the RRU, they are scheduled for transmission. If they are not sent during the scheduled subframe, they need to be discarded by the RRU. Due to this operation, UDP seems to be the only viable solution for measuring the fronthaul network overhead. Apart from the TBS data, information regarding the transmission is also sent, containing the scheduled subframe and physical resource blocks. The split is taking place upon the decision of the scheduler on which subframe the data will be sent (with the subframe duration being 1 msec), the modulation and coding scheme which will be used and the physical resource blocks that will be allocated for each UE.

Figure 2.5 is illustrating the performance results that are achieved for the MAC/PHY split. Due to the operation of this mechanism, we observe the bottleneck of the fronthaul link is around 1500 bits for TBS when using a 5MHz channel, and around 2000 bits for 10MHz channels. Throughput achieved by the LTE UE is around 14Mbps, whereas in the non-split case it was over 30Mbps. For both cases 5 and 10 MHz we can see that the fronthaul network reaches its capacity for MCS indexes over 14. From that point, the achieved throughput is less incremental, compared to the non-disaggregated framework.

2.6 Discussion and Future Work

In this work, we presented experimental results obtained through real experimentation and simulation, about the fronthaul performance of two different functional splits over the LTE protocol stack. We investigated the total delivered throughput to an LTE UE when the eNodeB process is running detached from an RRU, splitting the LTE networking stack at either the PDCP/RLC or MAC/PHY points. Our study concentrates on using existing technologies for the fronthaul network of Cloud-RANs, using 1Gbps copper links.

The results that we obtained illustrate that for high layer splits (i.e. PDCP/RLC), the transport protocols can pose performance limitations, but do not break the real-time operation of the base stations. Nevertheless, the results vary and as expected, stateless solutions (e.g. UDP) are found out to be more applicable. Moreover, for lower layer splits, like the MAC/PHY split, where the RRU transmissions are solely scheduled in the Cloud, real-time operation mandates the use of high bandwidth solutions, with the least possible overhead.

The proposed PDCP/RLC split can be used as a convergence sublayer among RRUs and BBUs that incorporate more than one heterogeneous wireless technologies. In the future, we foresee to investigate under real-world settings the impact of different functional splits in the low PHY layer, including the splits before the equalization of the signal at the receiver.

Chapter 3

Integration of heterogeneous wireless technologies in C-RAN

Contents

3.1 Chapter Introduction

Cloud-RAN is a key enabler for the 5th Generation of Mobile Networking systems (5G). The technology relies on decoupling the computational processing taking place traditionally on the base stations (BSs), and offloading part of it to Cloud instantiated VMs. The point at which the splitting of this process happens relies highly on the capacity of the fronthaul interface/technology used, between the Remote Radio Unit (RRU) and the Baseband processing Unit (BBU). Several studies have proposed different options for selecting the split point of the mobile stack (e.g. [24, 32]). Based on these, splits at the very low physical layer (e.g. I/Q samples are sent to the RRU), require constant high-rate connections between the BBU and the RRU, and hence costly fiber deployments. In such solutions, protocols such as Common Public Radio Interface (CPRI) or the newly introduced eCPRI are used for fronthauling the

RRU. Nevertheless, higher layer splits can be efficiently served over traditional Ethernet/IP connections for the fronthaul interface [114]. These higher-layer splits usually regard the splitting of the processing in legacy base stations either at the Layer 2 of the OSI stack or inside the PHY layer.

Cloud-RAN technologies facilitate the creation of multi-connectivity functional architectures for 5G systems. For example, the usage of a cloud-based convergence point consisting of joint processes for either MAC or PDCP layers and upwards is introduced in [91]. This trend is also reflected in the upcoming standards for the new 5G radio interface (e.g. [4], [5]), where a CU includes the processes of the PDCP layer and upwards, able to control multiple DUs incorporating the RLC layer and downwards. The communication between the CU and the DU is taking place over the newly introduced F1 interface, utilizing the F1 Application Protocol (F1AP), even supporting DUs providing heterogeneous wireless network connections (e.g. 5G, LTE, WiFi). A single CU should be able to serve multiple DUs (one-to-many relationship), whereas each DU is served from a single CU (one-to-one relationship). The data plane traffic (payload traffic forwarded to the network UEs) is traversed over the F1-U interface, encapsulating the traffic with GPRS Tunneling Protocol (GTP) headers over UDP/IP, similar to S1-U interface, whereas the control plane (e.g. RRC signaling) is using the F1-C interface, running over SCTP/IP, as in S1-C interface. Since this decoupling of the base station functionality takes place at a higher layer, it allows for lower layer splits to be also incorporated, thus creating a multi-tier disaggregated architecture. Therefore, we refer to this interface as the midhaul interface from this point onwards.

In this chapter, we build on top of our former work [68] and extend the PDCP/RLC data-plane splits with an organized protocol for managing the CU/DU communication. Using our protocol, we introduce WiFi-based DUs to the network managed through the same CU instance as the rest of the network. We employ the OpenAirInterface platform [83] for developing the splits over the implementation of the LTE protocol that is provided. On top of this functionality, we explore network performance for different transport protocols and latency settings on the midhaul link [71]. Through the utilization of the one-to-many relationship between the CU and DUs, we evaluate different policies for network selection/aggregation. We provide our experimental findings collected from a heterogeneous testbed offering all the components for our tests.

The rest of the chapter is organized as follows: Section 3.2 provides a literature overview of the field. Section 3.3 presents our protocol for the intercommunication between CUs and heterogeneous DUs, and our policies for network selection. Section 3.4 describes our experimental setup. In Section 3.5 we showcase our results and in Section 3.6 we conclude.

3.2 Related Work

Cloud-RAN is a key technology for 5G mobile networks, with multiple benefits for both operators and network users, as identified in several existing works (e.g. [109], [20]). All these works focus on identifying different splits, based on a high-throughput fiber interface for the network fronthaul, and splitting the stack before/at the baseband processing level. Nevertheless, multiple splits have been further identified, particularly within the higher layers of the networking stack, for example in Layer 1 or 2, which require lower capacity for the fronthaul interface. Examples of these splits are included in [24], [32]. Packet-based transferring of data over the fronthaul interface was initially introduced by China Mobile in 2015, through the Next Generation Fronthaul Interface (NGFI) [114]. In their white paper, six different splits that can be accommodated within an Ethernet-based fronthaul are analyzed, along with their requirements for serving remote units with different characteristics (e.g. number of antennas, resource blocks used, etc.). Similarly, the authors in [19], measure the impact of packetization for an NGFI based fronthaul interface and validate the transferring requirements for the identified splits. In [79], they discuss the processing overhead in a Cloud-based setup for similar split architectures.

The importance of the splitting processes is also pinpointed by the current 3GPP standardization efforts for 5G. In the current efforts for the new radio interface, the PDCP/RLC split is included in [4]. Based on this study, we adopt hereafter the 3GPP terminology and refer to the cloud-based units as Central Units (CUs), consisting from PDCP layer and upwards, and the remote radio units as Distributed Units (DUs), depicting the mobile networking stack from the RLC layer and downwards (see. Fig. 3.1). One of the PDCP roles in the mobile networking stack is to manage and rearrange the independent RLC entities. Thus, it may be used for the subsequent management of DUs (RLC and below layers) corresponding to different technologies, enabling higher network capacity and network selection policies even on per-packet basis, as shown in [57].

Incorporation of heterogeneous technologies on the mobile networking stack has been included in the 4G protocol standards as well. Through the introduction of the Xw interface, the PDCP instance of a 4G base station shall be able to communicate with WLAN based cell deployments, towards expanding the network capacity and utilizing the unlicensed bands [3]. This process is known as LTE-WLAN Aggregation (LWA) and utilizes the LWAAP protocol for the intercommunication and signaling of the different components.

In this work, we deal with the incorporation of this type of interface in 5G Cloud-RAN deployments. In our previous work [68], we provided an experimental evaluation of the PDCP/RLC and MAC/PHY splits. Our work has been developed in

OpenAirInterface [83], for supporting the (data plane) transferring of data. We concluded that the split that has the loosest requirements for the fronthaul/midhaul capacity is the PDCP/RLC. Based on our findings, and since the 3GPP standards are not yet finalized, we hereafter propose an architecture for the intercommunication of heterogeneous (LTE and WiFI) DUs with the respective CU. Based on this architecture, we propose and evaluate different policies to provide network selection for the downlink traffic (CU to DU communication) or use them for network aggregation.

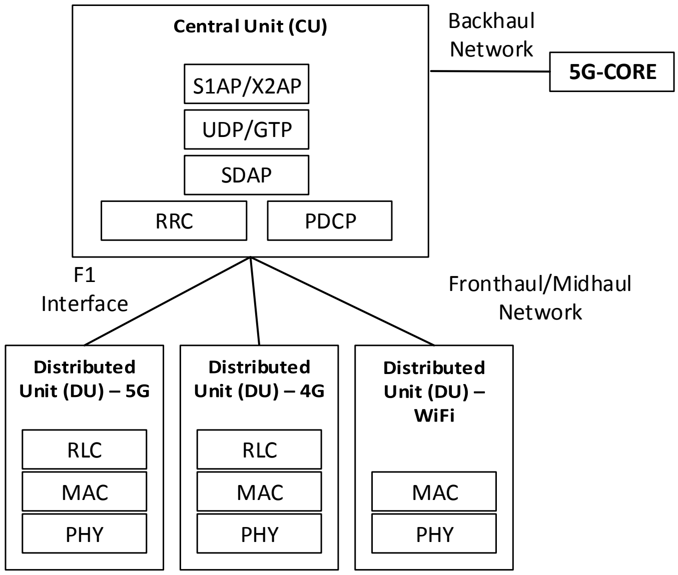


FIGURE 3.1: 5G Data Plane architecture for the CU/DU split

3.3 System Architecture

In this section, we describe the system architecture and the protocol that we developed for the intercommunication between the CU and the heterogeneous DUs. We use as a reference architecture the 5G RAN architecture (see Fig. 3.1) and as our implementation platform the OpenAirInterface platform. Hence, our reference networking stack is 4G stack, and we analyze the processes that take place in our developed functionality for the CU, the LTE DUs and the WiFi DUs.

3.3.1 Central Unit

The CU is incorporating all the processes from PDCP layer and upwards. Thus, it provides an interface to the Core Network (e.g. EPC or 5G-Core) for transferring GTP encapsulated user data. Moreover, it is integrating the RRC procedures for signaling and controlling the operation of the different layers (PDCP, RLC, MAC). In the 5G architecture, the Service Data Adaptation Protocol (SDAP) is introduced, as a means to map traffic flows to data radio bearers. In our system design, we intercept the data plane traffic only at the Service Access Point (SAP) interface with the

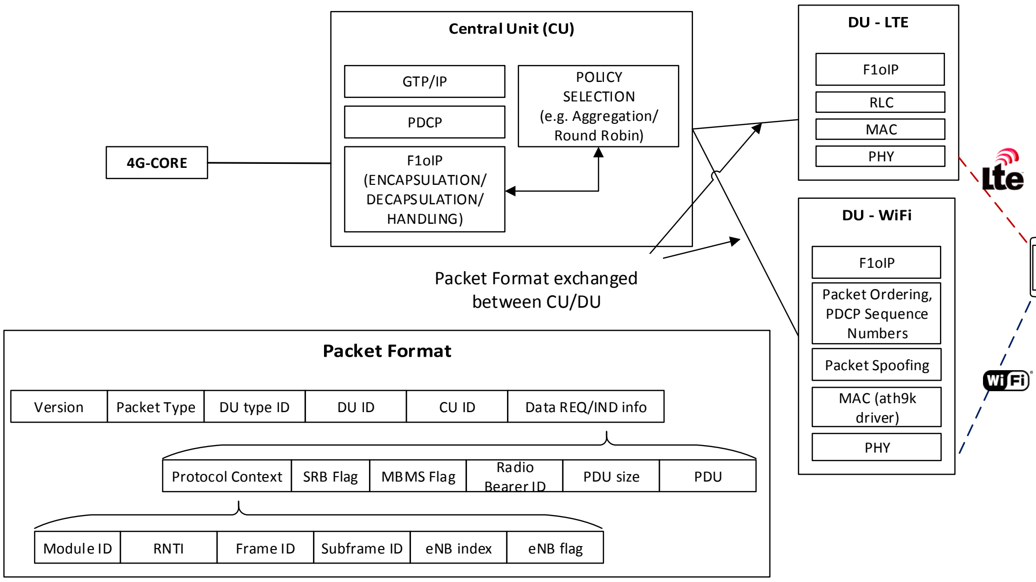


FIGURE 3.2: CU and DU split in OAI and the transported F1-U packet format

RLC. At that point, all the procedures of the PDCP layer have taken place, including stripping off the GTP headers, PDCP header encapsulation, numbering and data compression for the downlink (DL) traffic flow. The opposite procedures take place for the uplink (UL) traffic. Subsequently, our solution is encapsulating the PDCP PDUs with the appropriate headers to match the receiving DU(s) and forwards the traffic over the midhaul. This is the point where the network selection policies may be also running. The headers that we use are analyzed in subsection 3.3.4.

3.3.2 LTE Distributed Unit

The DU side of the architecture deals with receiving the F1-U packets over the midhaul interface, and based on the packet header is able to invoke the respective RLC processes. For the LTE stack that we use as reference for development, our protocol intercepts the data plane specific SAPs between the layers (*pdcp_rlc_data_req* for the DL flow and *rlc_pdcp_data_ind* for the UL flow) and subsequently is running the default processes provided by the stack. Therefore, if these processes need to return specific values on the further execution of the stack like e.g. the RLC Operation Status retrieved from the data request, we handle them by packing them to new messages and sending them back to the CU.

3.3.3 WiFi Distributed Unit

As the WiFi stack significantly differs from the mobile networking stack in terms of the supported procedures, different processes need to take place upon the reception of the data request for traversing the payload to the network UE or sending the data back to the CU. These processes include the reception of the data request transmitted from the CU, unpacking and stripping off the PDCP header, and subsequently delivering the payload to the wireless driver running on the DU device. For the UL data flow, payload traffic shall be encapsulated in the respective PDCP headers for the PDCP instance running on the CU. This includes dedicated processes for assigning new sequence numbers for the packets sent to the CU, as well as packet compression. For our implementation, and since the PDCP entity of OpenAirInterface is not supporting compression, we omit this procedure. In order to incorporate the information that does not exist in WiFi (e.g. protocol context, data bearer ID) and in order to allow the transparent handling of the packet reception at the CU side, we also introduce a lookup table at the DU side that is mapping the IP address of each receiving client to the protocol context information that we extract from incoming data requests. This process requires that the initial packet transmission happens from the CU to the DU, in order to keep this information. In the case where the end-client side uses a similar joint PDCP procedure, this process can be omitted.

3.3.4 Communication Protocol

The communication protocol that we employ is instrumenting the whole exchange process. We refer to our protocol and the procedures that take place as *F1 over IP (F1oIP)* hereafter. The CU and DU units are able to discover each other upon system startup, using a predefined capabilities and configuration file with the locations of the different modules. Upon the initial connection over the midhaul interface between the CU and the DUs, capabilities messages are exchanged with each other, stating the technology that is used by each DU. From this point, the exchange of the user-destined data taking place either on the DL or the UL channels is being carried out through our functions in the midhaul interface. Since we need to keep both ends informed of all the values needed for carrying out any computations at each receiving end (e.g. hash tables with the network users), we piggy-back the needed information in the packets that are exchanged.

The F1oIP packet format used is depicted in Fig. 3.2. Each PDCP data request/indication PDU is encapsulated in a packet including fields for packet type, DU type, and addressing the DU ID and the CU ID. Different types may be supported for the same DU, as a single unit may incorporate functionality for both technologies, whereas the selection of the interface is made by the CU. Fields containing the data request or indication information (Data REQ/IND info) are used for piggy-backing the information needed for carrying out computational functions at the different network ends. This information includes the protocol context, as well as the receiving UE RNTI, and scheduling information for the transmission over the air (frame/subframe). The overall overhead posed by this header, along with the current status in the size of the respective variables that are used and exchanged for OpenAirInterface is measured to be 80 bytes long. For the case of the WiFi-based DUs, this information is redundant and therefore ignored.

As an extension of the scheme, we foresee the incorporation of GTP tunnels for the communication (as drafted in the standards for 5G RAN). In such a setup, each GTP TEID is mapped to the respective variables for the UE, bearer IDs, etc. and therefore the context variables can be omitted.

3.3.5 Network Selection Policies

The decoupling of the base station stack to a CU/DU functionality, and the incorporation of heterogeneous DUs in the system, creates fertile ground for the application of network selection algorithms. Based on the output of these algorithms, the CU may select the DU to which the traffic will be forwarded to, and thus select which network will be utilized. Since all the traffic is sent over the PDCP layer, this selection can be performed even on a per-packet basis. As a proof-of-concept, we developed the following policies and further evaluate them in section 3.5:

- **LTE WiFi Aggregation**: In this mode, each data packet generated by the PDCP process is flooded to all available DUs in the system.
- **Round-Robin Scheduling**: For this policy, the CU selects to which DU to send the traffic in a Round-Robin manner.
- **Single Interface Selection**: This policy is forwarding traffic to only the selected DU.

Of course, these policies are only indicative. The system can be easily extended to host new policies for network selection, as well as gather information on the current network status and make decisions on the employed networks. This allows the implementation of traffic steering for aggregated networking topologies, e.g. selecting the transmission of time-critical data over LTE/5G and using WiFi otherwise.

3.4 Testbed Implementation

The described functionality has been developed in the OpenAirInterface platform and is executed over the NITOS testbed. NITOS is a heterogeneous testbed located at the premises of University of Thessaly, in Greece. It offers a very rich experimentation environment with resources spanning from commercial LTE, to WiFi and Software Defined Radio platforms that suit our experimentation needs [67].

For the development of the messaging exchange scheme, we employed Google's Protocol Buffers Library and the C language bindings [88]. By formatting the message header through the *protobuf* library, the overall header size of our communication solution, along with the piggy-backed information, is 80 bytes, that is exchanged between the CU and DU and vice-versa whenever a packet is transmitted over the network. The development of the CU/DU functionality has been written as a separate module inside the Layer 2 functionality of the OpenAirInterface code. As the transport protocol between the CU and DUs, we use an asynchronous TCP or UDP interface. The current configuration of the CU enables the utilization of different transport channels per each DU, thus allowing them to run with different settings (e.g. TCP for the LTE-DU and UDP for WiFi-DU).

The utilization of the protobuf library provides the opportunity for applications of different languages to use the same message definitions. Therefore, for the development of the WiFi DU, we used a Python-based agent. This agent is capable of receiving the CU messages, retrieving the payload and injecting it to the WiFi device that is configured as an Access Point. The injection is being handled by the scapy Python module [17], which provides bindings for creating packets and injecting them into a network interface.

The topology used for our experimentation process is given in Fig. 3.3. Since the current version of F1oIP is only overriding the data plane communication between the CU and the LTE DU, the production of two different binary files is not possible. However, we emulate this type of behavior by injecting delay between the network interfaces that are used for this communication between the CU and DU, equal to 0,250ms. The delay that we inject is done with the *netem* application and is equal to the mean delay that we measure over the midhaul between the CU and the WiFi DU.

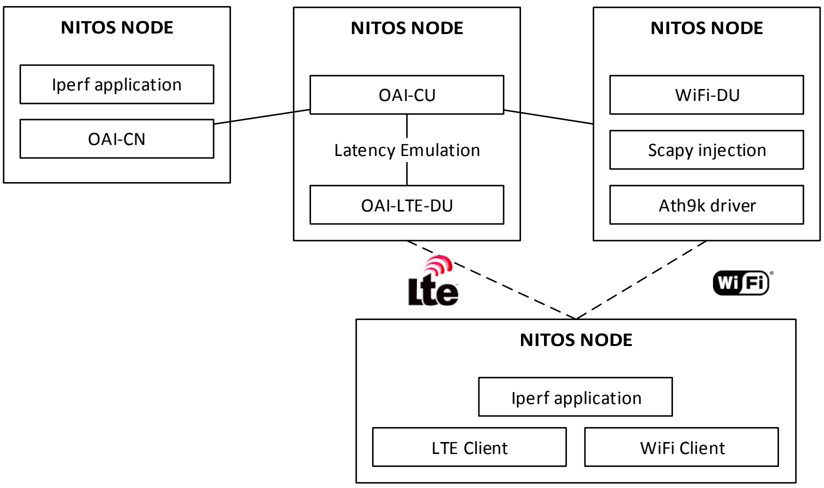


FIGURE 3.3: Experiment mapping over the NITOS testbed

In the following section, we provide our experiment results gathered from running the platform in the testbed. Each experiment is provided with a resolution of 10 for each measurement. For generating traffic for our measurements, we use the *iperf* traffic generator, set to saturate the wireless link with UDP traffic. The LTE and WiFi DU clients are always logged to use the same Modulation and Coding Scheme over the channel, for all the experimental measurements. The configuration of all the involved testbed components is provided in Table 3.1.

TABLE 3.1: Testbed Equipment parameters for experimenting with heterogeneous disaggregated Cloud-RANs

Network Parameters	Values
LTE mode	FDD Band 7
LTE Frequency	2680 MHz (DL)
Antenna Mode	SISO
No RBs	50 (10 Mhz)
UE	Cat. 4 LTE, Huawei E3272
Backhaul/Midhaul RTT	$\sim$ 200 msec
Backhaul/Midhaul capacity	1Gbps Ethernet
Ethernet MTU size	1400 bytes
WiFi Clients	Atheros AR9380

3.5 Experimental Results

The experimental evaluation of our scheme is organized in two subsections: 1) Initial benchmarking of the platform for the different policies for network selection, and in terms of Cloud resource consumption as the number of DUs increases, and 2) evaluation based on the delay over the midhaul.

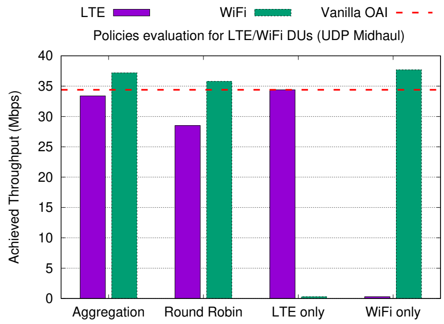
3.5.1 Policy Evaluation and Benchmarking

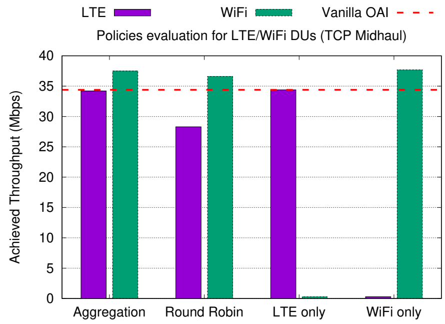
As a first set of experiments, we measure the performance of the network selection schemes listed in Section 3.3.5. We measure the single-unit *vanilla* OpenAirInterface eNB to achieve 34.4Mbps goodput for the DL channel for the under-test configuration. Subsequently we measure the performance of OpenAirInterface including our additions, for either UDP or TCP based midhaul (Figures 3.4a and 3.4b respectively).

We see that for the Aggregation mode, in which the CU is forwarding traffic to all available DUs, the achieved performance for the LTE network is close to the *vanilla* setup. Likewise, the single network selection policy produces similar results. This is due to the configuration of our protocol that exchanges signaling messages between the CU and the DU only during the initial setup phase. For the Round-Robin configuration we observed slightly lower performance for both DUs, caused by the extra delay induced in the system by the respective processes that determine the DU selection.

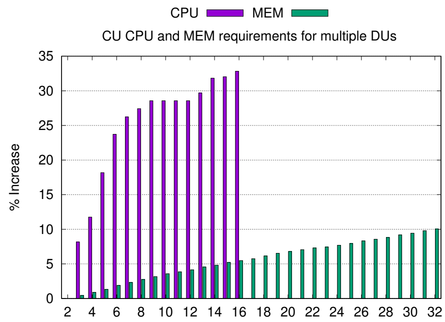
It is worth to mention here that the WiFi configuration is able to reach a maximum of 37.7 Mbps when saturating the channel. This limitation comes from the usage of the python Scapy module for injecting traffic to the WiFi interface; Scapy is opening a new socket connection for each new packet that arrives at the DU for delivering the traffic to the WiFi driver.

As a second benchmarking evaluation we measure the requirements for the CU in processing power and memory, when varying in the number of DUs deployed. Figure 3.4c depicts our experimental findings in terms of measured overhead for each





(A) Policy Evaluation using a UDP Midhaul (B) Policy Evaluation using a TCP Midhaul



(C) CPU/MEM allocation vs Number of DUs

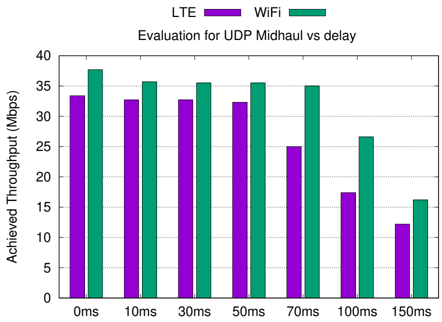
FIGURE 3.4: Experimental Evaluation for the developed policies

new DU introduced to the system, compared to an initial setup with 2 DUs. We use only WiFi DUs for this type of experiment. We measure the resource requirements for up to 16 DUs in the system, as at that point we determine that the CPU of the machine running the CU software is exhausted. As illustrated, the processing resources needed to run the CU for up to 8 DUs requires approximately 25% more processing power compared to the 2 DUs scenario. For supporting the remaining set of the DUs (up to 16 DUs) we require about 34% more processing power. Regarding memory usage, we observe an almost linear increase as new DUs are added to the system. Approximately, from the CU side, each new DU consumes additionally about 30MB of memory for its efficient operation.

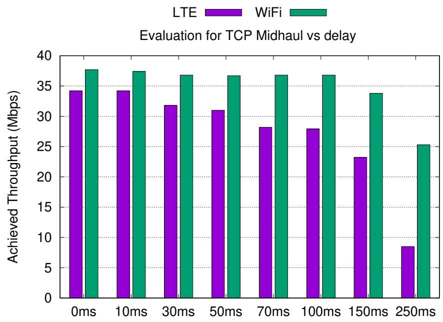
3.5.2 System Evaluation for varying Midhaul delay

As a second set of experiments, we measure the delivered goodput and Round Trip Time (RTT) for varying delay on the midhaul link. For the LTE case, we use the *socat* application to redirect the requests from the CU to an intermediary testbed node before delivering them to the DU. The latency on both the LTE and WiFi links is measured to be the same.

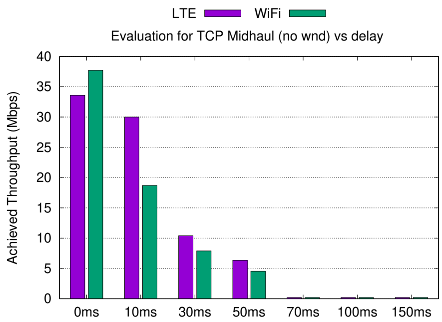
We use the *netem* application to set artificial delay on the midhaul link. We use the aggregation policy for these experiments, as this is the policy that produced higher results in the initial benchmarking experiments in the previous section. Figures 3.5a and 3.5b show the results for either UDP based midhaul or TCP. For both cases, we



(A) Evaluation for varying delay on UDP Midhaul



(B) Evaluation for varying delay on TCP Midhaul



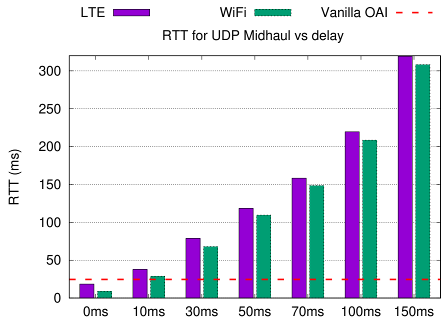
(C) Evaluation for varying delay on TCP Midhaul (no wnd)

FIGURE 3.5: Experimental Evaluation for varying Midhaul delay

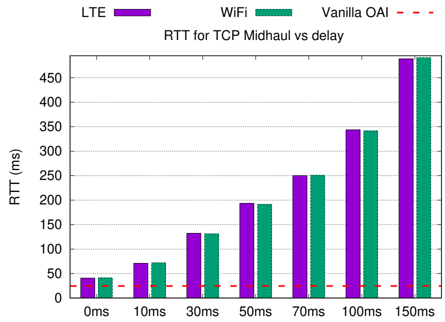
see that the performance starts to drop at around 70ms of midhaul latency. Nevertheless, the respective RTT (see Fig. 3.6) for the same interfaces seems to be growing by the double delay and a fixed amount added by the wireless access. Based on our results, we can incur that if the midhaul interface is realized over a fiber-based Ethernet link, the CU will be able to serve distributed DUs located at 500 Kms away without any decrease in the provisioned service at the end-client. Of course, in such environments, we need to further investigate on how to differentiate the paths that low latency applications take in order to minimize the impact on the user's QoE. UDP from that point and for higher delays, starts to perform worse than TCP, which through the adaptation of the congestion window and the receiving window is able to better handle the higher latency on the midhaul link. However, if these features are deactivated (see Figure 3.5c), we see that TCP cannot handle even lower delays in the midhaul (e.g. 30ms). For the WiFi case, we see that it is more resilient to delay, starting to drop for delays higher than 100ms. This is caused by the fact that the bottleneck of our implementation is not the midhaul interface, but the injection module at the WiFi DU.

3.6 Chapter Conclusion

In this work, we provided an experimental evaluation of a protocol enabling the Cloud-based convergence of heterogeneous networks, when operating with the CU/



(A) RTT Evaluation using a UDP Midhaul



(B) RTT Evaluation using a TCP Midhaul

FIGURE 3.6: RTT Evaluation results for different Midhaul interfaces

DU split. We detailed the communication protocol and the development process for the split operation of the OpenAirInterface networking stack. We provided proof-of-concept and performance experiments on the network selection policies that we use for the DUs, the indicative cost in processing power and memory allocation at the CU side, and the overall delivered goodput and RTT for varying delay over the midhaul.

Our results show that the proposed CU/DU split is not posing any performance limitations compared to the legacy eNB setup, but only adds up to the overall flexibility of the provisioned network. All our developments are publicly available through the OpenAirInterface repository currently merged in the master branch providing the data plane communication between CU and DUs. In the future, we foresee the incorporation of the RRC messaging as well in our communication scheme and the tailoring of the protocol according to the developments made to the 5G standards (e.g. incorporation of the F1AP protocol for the midhaul interface). We also plan the extension of the scheme to include network status messages, based on the network utilization (e.g. WiFi performance degradation due to external interference), and the subsequent management of the involved DUs from the CU point. Finally, we expect the incorporation of pricing schemes, located at the CU for the DU selection, according to selections made by the network's end-users.

Chapter 4

Spectrum Coordination in heterogeneous C-RANs

Contents

4.1 Chapter Introduction

The incorporation of heterogeneous technologies at the end-user access has the potential to increase the overall offered network capacity, and allow operators to further expand their coverage through low-cost solutions, such as WiFi. This integration of heterogeneous technologies to the traditional cellular infrastructure has been approached through different manners in the past [2] in two manners: 1) by offloading the cellular traffic to WiFi networks, and 2) by aggregating these networks from either the Core Network side or the base station side, and offering multiple seamless connections to the serviced User Equipment (UE). As a matter of fact, traffic offloaded to hotspots or femtocells has surpassed the overall traffic transmitted to the Internet compared to traditional base station units [21], highlighting this paradigm shift from traditional macro-cell based setups to ultra-dense heterogeneous networks.

The 5th Generation of mobile networks (5G) is expected to boost existing network flexibility in terms of management and control of the edge access nodes, through the disaggregation of the base station units and their instantiation in the Cloud. Via Cloud-based Radio Access Networks (Cloud-RAN), base stations can be instantiated on the fly in an area, based on the demand that is perceived by the operator. Several points of disaggregating the base station stack have been proposed in literature, yet the scheme that is currently standardized for the 5G New Radio interface (5G-NR) regards the disaggregation of the base stations between the higher Layer 2 of the cellular network stack, between the Packet Data Convergence Protocol (PDCP) and the Radio Link Control (RLC) layers. In this split, Cloud-located unit as annotated as a Central Unit (CU) and the radio elements as Distributed Units (DUs) [6]. CUs incorporate the functionality of the layers from the PDCP layer and upwards, whereas DUs the functionality of the RLC layer and downwards. The selection of this point of splitting the stack is of major importance: it allows the incorporation of other lower layer splits inside the DU, thus transforming the CU-DU link as a Midhaul interface, whereas it allows several technologies to be aggregated through a signal point at the base station level inside the PDCP layer. Similar aggregation of technologies took place for the legacy LTE technology as well, with the incorporation of the LTE-WiFi Aggregation Adaptation Protocol (LWAAP) [3]. In the context of 5G-NR, the technologies that can be aggregated regard 5G-NR, legacy LTE and WiFi, as shown in Figure 4.1. This is the architecture also that we consider for this chapter.

As the deployments of different technologies in an area become denser, the available wireless spectrum crucial for their performance becomes more scarce. Especially when considering heterogeneous technologies in the RAN, efficient coordination is required in order to achieve spectral efficiency in a given area. In this chapter, we deal with the proposed CU/DU split of the base station stack, integrated with non-3GPP technologies, based on our prior contributions in [65]. In such a setup, our contributions are the following:

- To provide new signaling for collecting usage statistics of the heterogeneous technologies that are available in the area.
- To introduce and apply algorithms handling the spectral overlap of the different RANs, in order to efficiently place the different cells in the available frequency space.
- To experimentally evaluate the added functionality, demonstrate and prove its efficiency.

We use the OpenAirInterface (OAI) platform [83] as our development platform, and we evaluate our algorithms under real network settings in a testbed setup. Our results showcase efficient allocation of the under-study networks within a single congested wireless band. The remaining of the chapter is organized as follows: Section 4.2 is providing some former background and motivation for our work. Section

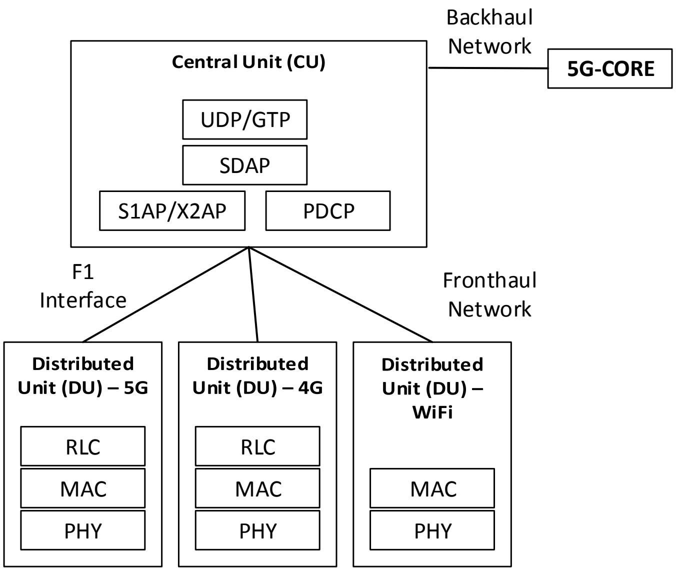


FIGURE 4.1: Disaggregated Base Station architecture according to the 5G-NR specification for F1AP [6]; multiple heterogeneous and legacy DUs can be managed through a single CU instance, located at an Edge data-center.

4.3 provides information on the system setup, relevant signaling used for collecting network statistics and the development of algorithms supporting the coordination of the disaggregated distributed units. Section 4.4 presents our experimentation platform and topology used for our experiments. Section 4.5 showcases our experimental findings, whereas in Section 4.6 we conclude our work.

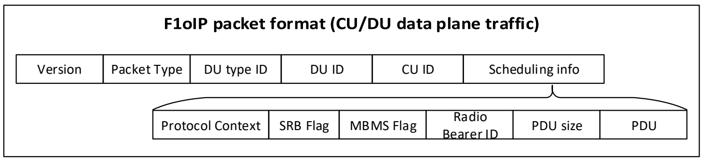
4.2 Related Work

Base station disaggregation has been an extensively investigated topic in relevant literature, as the realization of Cloud-RANs may potentially yield several benefits for both network operators and users. Works [109] and [20] indicate possible points of disaggregation of the base station and analyze the benefits from the operator's point of view. However, these works assume a high-throughput low-latency fronthaul link between the disaggregated base station components. Similar splits have been in higher layers of the base station stack, which can be efficiently served through a packetized fronthaul interface, with lower demands for latency and capacity. The disaggregation of base stations between the PDCP and RLC layers has been included in the 3GPP standardization of 5G New Radio (NR), through the introduction of the F1 Application Protocol (F1AP) [6]. The F1AP is the protocol for the packetized intercommunication between the Central Units (CUs) integrating PDCP and above layers, and the Distributed Units (DUs) of the network. F1AP has two variations: F1-U for transferring the user plane traffic over tunnels through the GPRS Tunneling Protocol (GTP), and F1-C for the control plane traffic, running over SCTP. From the CU point of view, the connections that can be maintained with the DUs is $1 : n$, meaning that each CU may control multiple DUs, whereas from the DU point of view is 1 : 1, meaning that each DU can be controlled by a single CU.

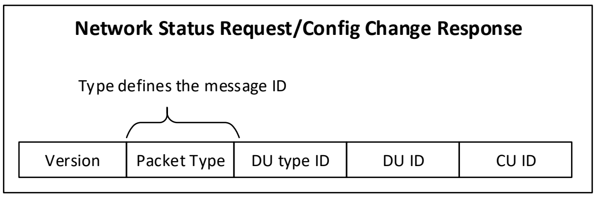
In the proposed architecture for disaggregated Cloud-RANs, support for heterogeneous DUs has been incorporated, as shown in Figure 4.1. The point of disaggregation is very convenient, as one of the PDCP roles in the mobile networking stack is to manage and rearrange the independent RLC entities. Thus, it may be used for the subsequent management of DUs corresponding to different technologies, enabling higher network capacity and network selection policies even on a per-packet basis, as shown in [57]. Incorporation of heterogeneous technologies on the mobile networking stack was included in the 4G protocol standards as well. Through the introduction of the Xw interface, the PDCP instance of an LTE base station is able to communicate with WLAN based cell deployments, towards expanding the network capacity and utilizing the unlicensed bands [3]. This process is known as LTE-WLAN Aggregation (LWA) and uses the LWAAP protocol for the intercommunication and signaling of the different components.

Nevertheless, aggregation of multiple heterogeneous links within a single area may entail performance degradation in the cases of overlapping spectrum usage. This is mainly an issue for WiFi-like technologies that use the CSMA/CA mechanism for accessing the wireless medium, and hence apart from interference they are subject to contention with other neighboring cells operating in the same spectral area. Hence, an efficient coordination mechanism for the different technologies used to provide data services to the users is needed. In [64], the authors argue the applicability of different coordination mechanisms for including heterogeneous networks in the enhanced Intercell Interference Coordination (e-ICIC) mechanism that LTE networks may implement. Similarly, in [18] authors observe the coexistence between WiFi and LTE within the same spectrum, paved by the suggestions for enabling LTE to opportunistically access the unlicensed bands in order to increase the channel capacity [39]. The authors introduce through eICIC a novel coordination scheme as a coexistence solution.

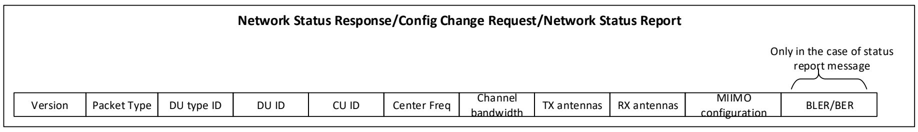
In this chapter, we build on top of a real Cloud-RAN setup, in order to introduce coordination functions for heterogeneous technologies. We use the OpenAirInterface [83] platform for our developments, and make use of the extensions to the platform [65] that introduce heterogeneous wireless technologies to the RAN, controlled by a single point located at the CU side of the base station. The implementation makes use of dedicated signaling between the CU and heterogeneous DUs, introduced as F1-over-IP (F1oIP) due to its resemblance with the F1AP protocol. We extend the scheme to introduce new signaling between the DUs and the CU, in order to retrieve the appropriate RAN configuration settings and conclude on the optimal use of spectrum in an area. The following section details the developed signaling between the CU and DUs.



(A) Packet format for data plane traffic exchanged between CU and DUs



(B) Network status request (CU to DU) and config change response (DU to CU) message format.



(C) Network status response/report (DU to CU) and config change request (CU to DU) packet format]

FIGURE 4.2: Message format exchanged between the CU and DUs regarding data plane traffic and spectrum coordination messages.

4.3 System Architecture

In this section, we detail the existing signaling between the disaggregated heterogeneous base stations, and the extensions to support the heterogeneous cell coordination in terms of spectrum. We use as our starting point the F1oIP implementation for the disaggregation of base stations as CUs and DUs, detailed in [65], and extend it accordingly in order to enable the spectrum coordination between the heterogeneous entities of the network. The implementation is introducing a new signaling mode between the PDCP and RLC layers, resembling the F1AP standardized interface for the communication between CUs and DUs. Originally, F1AP is handling data plane packets over GTP tunnels, established for each served UE of the system. The F1oIP implementation is using UDP/TCP interfaces in order to exchange the traffic between the CUs and DUs, including also some signaling information on the packet headers that is ordinarily exchanged between the PDCP and RLC layers (e.g. DRB/SRB allocation, frame/subframe scheduling, protocol context, etc.). The following sections initially describe the new packet format introduced with the F1oIP messaging mechanism and the roles of the CUs and DUs, and later on, we introduce the new messages for facilitating the coordination.

4.3.1 Disaggregated base station communication

According to 5G-NR specifications, the disaggregated functionality of the 5G base stations shall address several technologies. To this aim, the proposed standardized split option by 3GPP resides in the high layer 2 of the OSI stack, between PDCP and

RLC. As this split option has more slack limitations on the latency and throughput over the fronthaul [114], different technologies can be used in the wireless part of the DU. Target network access technologies offered by the DUs are 5G-NR, LTE, WiFi, and their evolution.

We use as a starting point the implementation of F1oIP, which handles the communication between the CU and heterogeneous DUs of the system. In this implementation, each packet exchanged over the fronthaul interface is bearing on its header scheduling information to be used by the lower layers, according to Figure 4.2a. Based on this information, the packet is assigned to the respective transport channels of RLC and is then left to the MAC layer for scheduling its transmission over the air. For the case of non-cellular DUs, the respective information is not handled from the respective DU software. For example, in the case of a WiFi DU, the F1oIP header information related to the scheduling of the packet is ignored. For the UL case, the reverse process takes place before transmitting the packet to the CU. This means that the DU is assigning new PDCP sequence numbers and creates the respective header in order for the packet to be handled at the CU side.

4.3.2 Coordination Messages

In order to enable central management of the heterogeneous DUs, we introduce some new signaling messages used for exchanging the capabilities and network configuration between the CU and DUs of the network. To this aim, we implement a second communication channel between the CU and DUs, apart from the data channel, that is exchanging this type of messages. The messages that we introduce are:

- The **Network Status Request** message, sent by the CU and requesting from a single DU about its current RAN configuration. The message format is shown in Figure 4.2b.
- The **Network Status Response** message, sent by the DU responding to a Network Status Request message. The format of the message is shown in Figure 4.2c and includes the configuration on the center frequency, channel bandwidth used, number of antennas used, and MIMO configuration.
- The **Configuration Change Request** message, sent from the CU after the coordination algorithm has taken place and has concluded on the new spectrum allocation of the network. The message format is the same as the Network Status Response message.
- The **Configuration Change Response** message, sent from the DUs of the network as an acknowledgment that the new configuration advertised to the DU has been applied to the RAN.

- The **Network Status Report** message, sent periodically by the DUs to their managing CU as a keep-alive message. The message contains similar fields as the Network Status Response message but contains also information on the error rates that are perceived over the network from the DU side. Hence, the reception of such a message where a DU is reporting several losses over the network may trigger the spectrum coordination algorithm at the CU side.

In order to enable the exchange of these messages, a respective agent message has been written on the DU side in charge of synthesizing and parsing these messages and issuing the appropriate commands. As we target the coordination of WiFi-based DUs, complementary to the cellular-based DUs (5G-NR and LTE), we make use of the Channel Switch Announcement (CSA) feature that WiFi incorporates for imminent channel changes. This allows us to reconfigure the cell and the associated channels, based on the information of dedicated WiFi messages indicating a channel switch after a number of milliseconds.

4.3.3 Coordination Service and Algorithms

Based on the above messages, the coordination service for the heterogeneous DUs is summarized in Figure 4.3. The coordination system is based on the client-server model, with the server-side being located at the DU side of the communication channels. Upon system startup, the CU that has information of the managing DUs, sets up this coordination communication channel with all the DUs and starts to periodically query all the DUs for their current wireless configuration. Upon the reception of such a message, all the DUs respond with their settings. Once the CU has the information collected from all the DUs, checks whether there is any overlap in the frequencies. If so, an algorithm determining the new frequencies is executed, and *configuration_change_request* messages are sent to the DUs.

It is worth to mention that the requests may include new configurations for the antenna configuration of the DU, the placement of the secondary channel in the case of an IEEE 802.11ac/n access point, and even the configuration of the channel bandwidth using the methods indicated in [56]. The algorithm that is running on the CU is distinguishing two different types of coordination: 1) coordination for heterogeneous technologies, e.g. in the case of an LTE DU operating in unlicensed spectrum [39] in the same frequency spectrum as a WiFi DU, and 2) in the case of homogeneous technologies, e.g. only LTE/5G-NR or only WiFi. For the former case, we use the incentives that are provided in [70], where the authors examine the coexistence of LTE and WiFi cells. Based on their conclusions, in the case of overlapping or partially overlapping configurations, we move all the WiFi traffic by at least 2.5MHz away from the cellular traffic, in order to mitigate any performance issues. For the case of homogeneous cells, we use the approach highlighted in [90], which uses a graph coloring approach to determine the WiFi cells that are operating in an area.

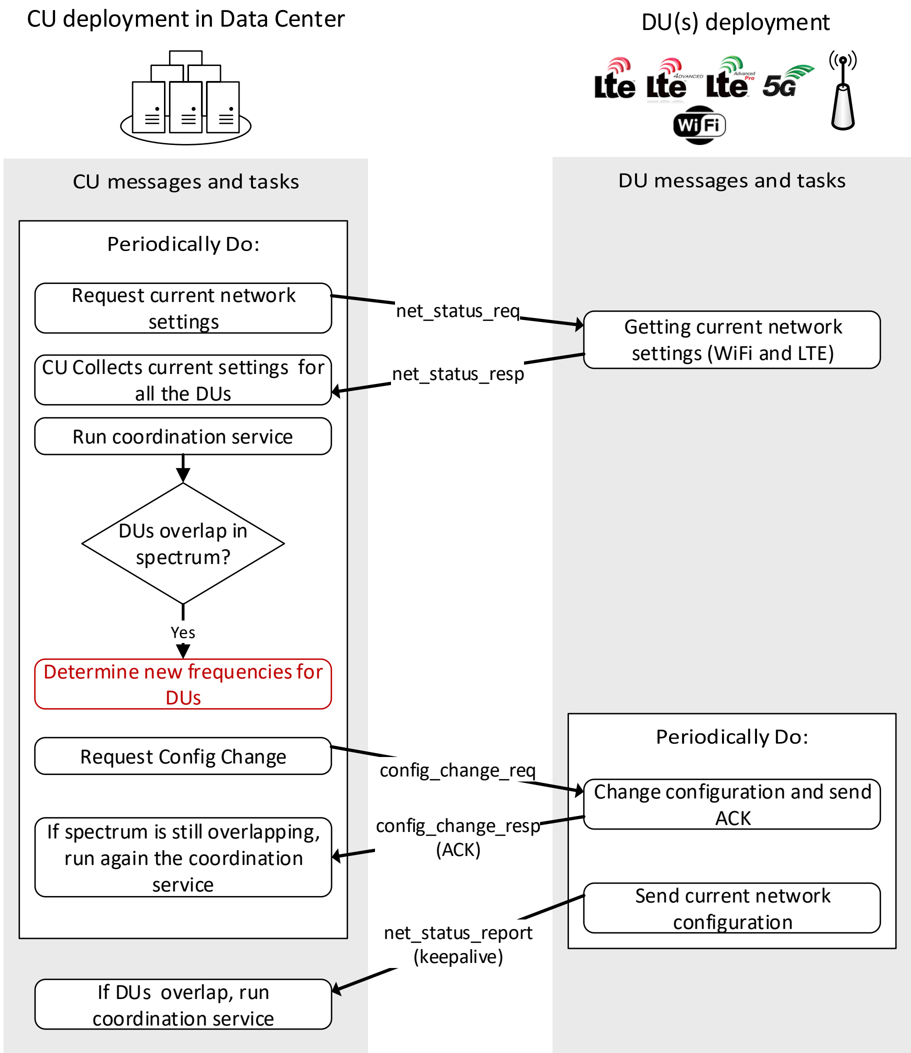


FIGURE 4.3: Signalling and process developed for coordinating heterogeneous DUs managed from a single CU. The coordination function takes place at the CU side and the new configurations are transmitted to the DUs.

Our algorithm begins with examining the case of heterogeneous technologies overlapping spectrum, and then subsequently checks for any homogeneous technologies overlapping case. These processes are taking place within the red-colored process at the CU, indicated in Figure 4.3.

4.4 Testbed Setup

As our development platform for the described functionality we have been using the OpenAirInterface [83] platform, which provides a software based full stack implementation of contemporary cellular networks. We evaluate the framework in the NITOS testbed, an open and remotely accessible infrastructure located in University of Thessaly, in Greece [67]. NITOS is offering a wide selection of resources, spanning from commercial LTE to open-source WiFi and several Software Defined Radio

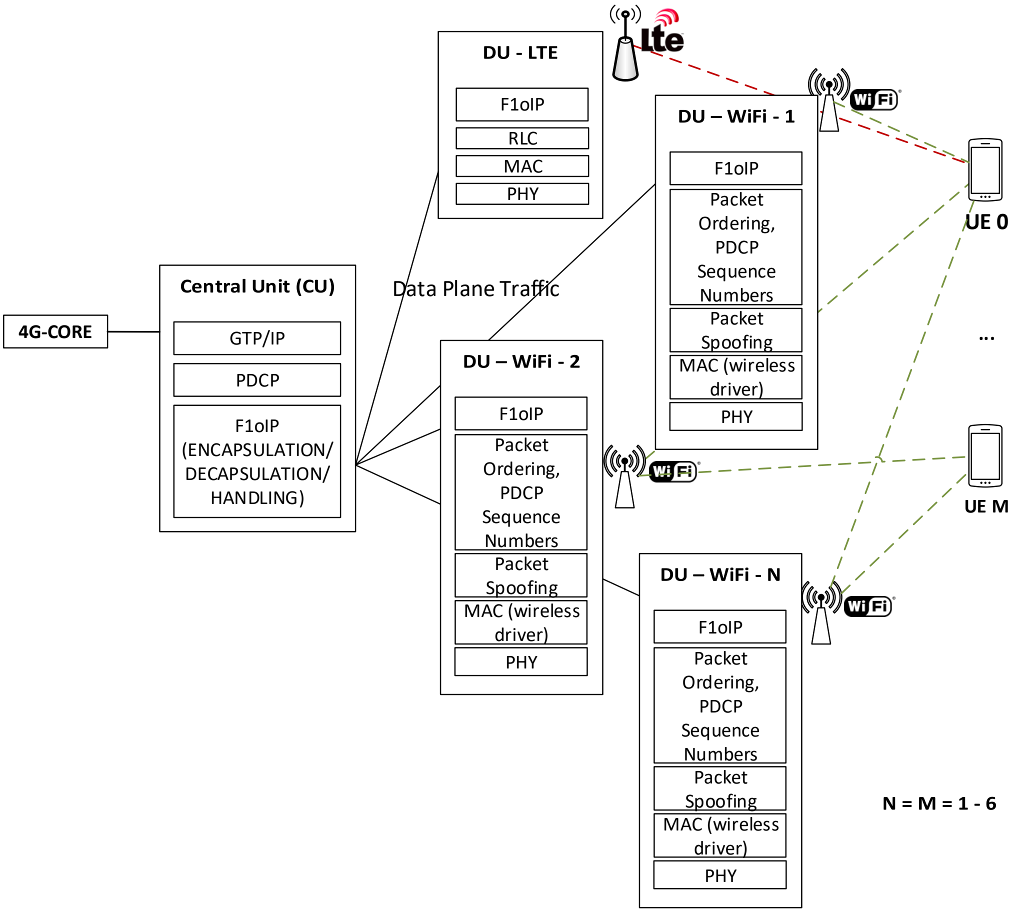
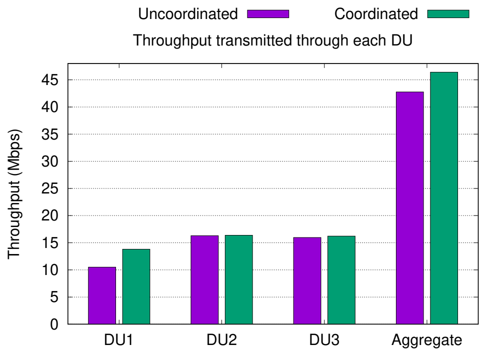


FIGURE 4.4: Experimental setup used for evaluating our scheme

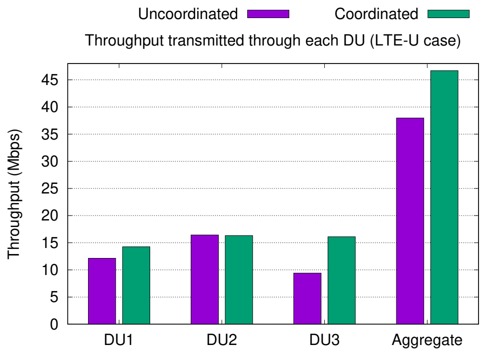
devices, used to set up our experimentation system. Figure 4.4 presents our experimental setup, with the F1oIP framework [65] being setup at all the WiFi and LTE DUs of the system. Although the OpenAirInterface platform currently supports both LTE and 5G-NR, the developments in the latter technology only span the physical layer. Hence, our experiments are solely limited to LTE and WiFi DUs.

For the development of the respective messaging scheme, we used Google's Protocol Buffers Library. The *protobuf* library allows us to create the messages in different languages, using the same message definitions. Hence, the implementation of the WiFi DU agent is a Python-based, with two different parts; one for receiving the data plane messages from the CU side, and injecting them to the WiFi network or receiving them and packaging them accordingly to send them to the CU, and a second part in charge of collecting the statistics and exchanging the coordination related messages with the DU.

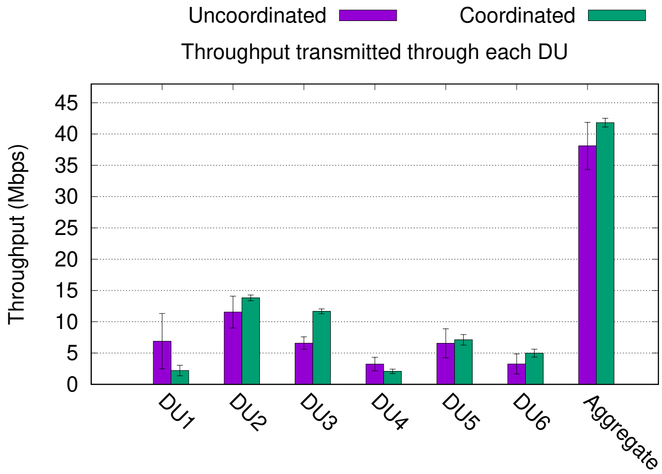
The topology used for our experimentation process is given in Figure 4.4. As the split for OpenAirInterface regards only the data plane operation of the platform CU and LTE DU are collocated on the same service. However, we emulate the disaggregated type of behavior by injecting delay between the network interfaces that are used for the F1oIP communication between the CU and DU, in the range of 0,250ms. The



(A) Throughput achieved over each DU for 3 WiFi DUs



(B) Throughput achieved over each DU for 3 WiFi DUs coexisting with an LTE cell



(C) Throughput achieved over each DU for 6 WiFi DUs

FIGURE 4.5: Experimental results for 3 - 6 WiFi DUs within the same band: Overall throughput exchanged over each DU is depicted

delay that we inject is done with the *netem* application and is equal to the mean delay that we measure over the interface between the CU and the WiFi DU. The number of WiFi DUs is ranging from 1 to 20, thus creating a highly dense heterogeneous network consisting of WiFi and LTE devices. Since none of the nodes is able to bear more than four heterogeneous technologies for setting it up as the receiving UE, we employ different nodes for the WiFi-based UEs. By using the extensions built in [65], we inject traffic from the Core Network side an the CU is broadcasting this information to all the DUs in our system, thus creating a high contention system. We use the *iperf* application to saturate the network with UDP traffic, and measure the delivered traffic at each node (throughput) over each DU. The following section presents our experimental findings for ranging the number of different DUs in the network.

4.5 System Evaluation

In this section, we provide a proof-of-concept evaluation of our scheme. We organize our experiments in the following manner: 1) initially we conduct experiments using the 2.4GHz band and place a contending LTE-U cell within the band and 2) subsequently we place up to 6 different WiFi DUs within the same band, that create an environment high in contention and interference. The depicted results are averaged from 10 different experiment runs in the testbed.

For the first set of experiments, we use three different WiFi DUs and an LTE DU. We compare our algorithm with the default process that is running at the WiFi driver of the DU for auto-selecting the transmission channel. Figure 4.5a shows our results. For the uncoordinated case (each WiFi driver selects its own channel), all the WiFi DUs are selecting their transmission channel to be one of the non-overlapping WiFi channels in the band (channels 1, 6 and 11). For most of the cases that the experiment was executed, two DUs were automatically switched to operate in the same channel, thus suffering from contention. Our CU co-ordinated case is assigning channels in a similar manner, but also takes into account the current allocation of other DUs in the same area, and thus is assigning different channels to them. This solution suffers from interference, but in the case of low numbers of DUs in the area, this problem is mitigated. This is reflected in the measured aggregate throughput, where our coordinated solution is achieving more than 10% higher overall throughput.

Figures 4.5b shows the case where at channel 6 of the band we set up an LTE DU to operate. In this case, our solution is able to retrieve the setup of the LTE DU and coordinate all the WiFi DUs to organize in a manner that they are under no destructive interference from the LTE cell. For such a case, and for the 3 WiFi DUs that we used, we see that the aggregate throughput is enhanced by almost 20%.

Figure 4.5c depicts our results when using up to 6 different WiFi DUs, located within the same frequency band in overlapping frequencies. We observe that the autoselect channel feature results in higher throughput being delivered over some DUs, but the measured values present high deviations between different experiment runs. It is indicative that DU1 and DU4 manage to deliver through different experiment runs constantly lower throughput, however, the deviation in the provided measurements is very low. The lower throughput is imposed due to the interference created by all the other DUs operating in overlapping channels. Nevertheless, the coordination algorithm is able to deliver higher aggregate throughput over the setup and with lower deviations from the mean values across different experiment runs. The provided evaluation showcases experiments where the DUs are reorganized only in terms of spectrum. However, the developed protocol allows the organization based on the physical characteristics of the DUs (e.g. number of antennas, placing secondary channels, etc.) and a more sophisticated algorithm may yield better results.

4.6 Chapter Conclusion

In this chapter, we presented a protocol for the coordination of DUs in ultra-dense heterogeneous network setups. We detailed the developed signaling over the OpenAirInterface platform for the communication between a single CU and multiple heterogeneous DUs, and applied an algorithm for the selection of the optimal operating frequencies as a proof-of-concept experiment. Our experiments targeted the highly congested 2.4 GHz band and employed WiFi operating DUs. The results denote that the coordination may be used to efficiently provide higher overall capacity while showing lower throughput volatility for different experiment rounds.

In the future, we foresee the further development of spectrum coordination algorithms from the CU point of view. As the developed protocol is allowing the collection of physical interface characteristics in the CU side, more sophisticated algorithms can be developed that allow the reconfiguration of the DUs regarding the number of antennas, transmission power, etc. Moreover, we will seek to investigate how the allocation of clients at each DU may be perceived at the CU side and introduce signaling for the allocation of the clients to DUs that can serve them based on the overall energy efficiency of the deployed infrastructure.

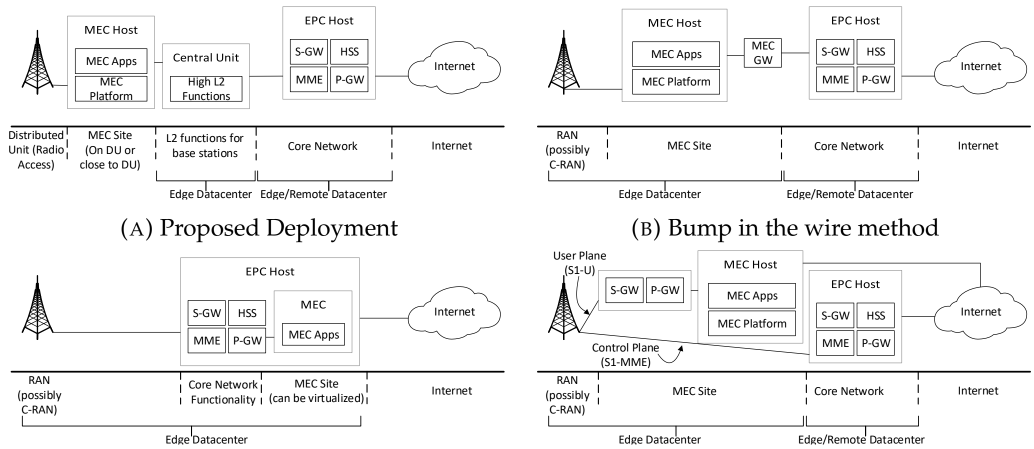
Chapter 5

Multi-access Edge Computing in disaggregated 5G base stations

Contents

5.1 Chapter Introduction

The fifth generation of mobile networking (5G) is fostering several advancements in the access, edge and core network, promising to offer significantly higher network capacity with lower latency over the network, allowing a variety of applications and critical services to thrive around this ecosystem. 5G brings advancements through the wide application of the Multi-access Edge Computing (MEC) concept [78], formerly annotated as Mobile Edge Computing, realizing the low-latency UE to service connections needed for the delivery of time-critical data. In the concept of Multi-access Edge Computing, heterogeneous technologies reside in the user access network, adding up to the overall capacity of the network through the formation of Ultra-Dense Networks.



(A) Proposed Deployment

(B) Bump in the wire method

(C) Collocation with EPC

(D) Local Breakout mode

FIGURE 5.1: Proposed MEC deployments vs existing deployments [38]

The advent of new network architectures, like the disaggregation of base station units based on the Cloud-RAN concept [68], allows the re-conception of technology paradigms that have been around since legacy LTE technology, like MEC. 5G-NR specifications [4] include the disaggregation of the base stations based on the 3GPP Option-2 split [1], between the Packet Data Convergence Protocol (PDCP) and Radio Link Control (RLC) layers of the mobile stack. This creates two different entities for a single base station: the Central Unit (CU), located at the edge cloud, and the Distributed Unit (DU), providing the actual access network to UEs. DUs can be of different access technology (5G NR, legacy LTE or even WiFi), served through a single operator's CU and Core Network. Each CU may be associated with multiple DUs, handling how each UE is served through the network (e.g. through aggregation of technologies or splitting the traffic over multiple DUs), whereas from the DU point of view each unit is associated with one CU.

Although MEC has been extensively considered as a low-latency solution since the initial drafting of 5G, its application is merely mapped to the new network architecture; for instance, for the case of the *bump-in-the-wire* [38] method in legacy setups (LTE), S1 messages are intercepted between the eNB and the Core Network. For the 5G-NR case, MEC servers are just collocated with the cloud-part of the disaggregated base stations, providing exactly the same functionality. However, the new network architecture allows the consideration of placing the MEC service deeper in the network, closer to the network edge. In this chapter, we propose the placement of the MEC servers closer to the network edge, collocating them possibly with the DUs of a disaggregated base station setup, realizing truly the Edge Computing concepts. This is, to the best of our knowledge, one of the first attempts to do so, and bring computational devices close to the true edge of the network. We extend the initial prototype presented in [66], and further develop it in order to:

- Include new signaling for supporting multiple applications concurrently with multiple users.
- Couple it with a multi-technology base station allowing seamless switch of wireless technologies serving the end-users.
- Provide a fast switching solution for selecting the technology through which each end user will be served, in order to minimize UE-to-service and vice-versa latency.
- Demonstrate the performance gains in terms of latency and Quality of Experience when placing MEC services close to the DU, contrary to other approaches suggested in relevant literature.

The rest of the chapter is organized as follows: Section 5.2 provides a brief literature overview of the application of MEC in 5G. Section 5.3 provides the extensions of the framework in terms of signaling, and how we are able to select the technology through which each network UE will be served. Section 5.4 provides our testbed setup for the evaluation of our solutions, and Section 5.5 shows our experimental findings. In Section 5.6 we discuss our findings and possible applications, whereas in Section 5.7 we conclude the chapter and present some future directions.

5.2 Motivation and Related Work

The integration of computing resources, closer to the network edge, has been a key research issue since the legacy LTE protocol, with interfaces being lately developed specifically for integrating the MEC functionality in the 5G context [29], [54]. The different methods for deploying the MEC services and applications have been discussed in [78] and [38], providing some guidelines for the maximum latency for the UE to service and vice-versa path for some state-of-the-art 5G applications (e.g. Industry 4.0, eHealth, AR/VR, etc.). These deployments are summarized in the following (see Figure 5.1):

1. The *bump-in-the-wire* method, where the MEC service is placed on the backhaul link of the base station, interconnecting it with the Core Network and intercepting the data plane traffic of the cellular network and redirecting it to the MEC applications
2. Collocating the core network and the MEC servers at an Edge datacenter. In such a case, IP traffic is only intercepted beyond the PDN-GW component and redirected to the MEC applications.
3. Using a distributed core with a *Local Breakout (LBO)* mode, with control plane traffic being redirected to another core network instance than the data plane. The MEC services are introduced closer to the edge, co-located with the core network instance handling the data plane traffic.

The MEC applications/services can also be virtualized, allowing further innovations to take place, such as their live migration to data centers located closer to the UEs, based on their trajectory, etc.

Through the introduction of Cloud-RAN based networking, the mobile networking stack has been redefined and disaggregated at different levels. From the variety of different splits proposed for the stack [20], the split between the PDCP and RLC layers has been standardized in the 5G New Radio (NR) specifications [6]. These specifications define the Central Unit (CU) that integrates the processes of PDCP and above layers and can be instantiated at an Edge Datacenter in order to manage one or multiple Distributed Units (DUs), that may be of heterogeneous type, and integrate the processes of the RLC layers and below. The communication between the two entities takes place through the F1 Application Protocol (F1AP), over the Stream Control Transmission Protocol (SCTP) for control plane traffic (F1-C), and over the GPRS Tunneling Protocol (GTP) over UDP for the data plane traffic (F1-U). Moreover, the disaggregation of the base stations in the PDCP layer allows multiple technologies to be integrated in the network cell; after all, the operation of protocols such as the LTE WiFi Aggregation Adaptation Protocol (LWAAP) [3] considers the control of WiFi cells from the PDCP instance of the base station. This functionality is also drafted in the current standardization activities for 5G-NR, providing hooks for the integration of non-3GPP technologies and legacy technologies to the 5G-NR disaggregated cell [6].

Despite the disaggregation of base stations, possible MEC deployments do not consider moving the edge services closer to the DU; in the best case, the services are co-located with the CU at the edge datacenter [38] (case illustrated in Figure 5.1b). The F1-U interface, that carries the user plane data, is based on GTP encapsulation and is very similar to the S1 Application Protocol (S1AP) for communicating between the core network and the CU. Thus, engineering a solution for bringing the services at the true edge of the network and closer to the DU should not pose a big overhead for technology providers. In our work, we propose moving the MEC services closer to the true network edge, based on the prototype built in [65] that introduces multi-technology base stations and in [66], that introduces services on the fronthaul interface.

In [65], we provided a prototype for experimenting in disaggregated heterogeneous 5G architectures, integrating non-3GPP technologies (WiFi) on the same CU. This prototype provided proof-of-concept experiments determining the maximum distance between the CU and heterogeneous DUs so as no service disruption is experienced at the UE side. This implementation used TCP/IP channels for the data plane communication between the PDCP and RLC layers of the stack, providing dedicated signaling for this purpose. This signaling is referred as **F1 over IP (F1oIP)**, as it has a similar structure with the standardized F1AP [6]. Since these splits use

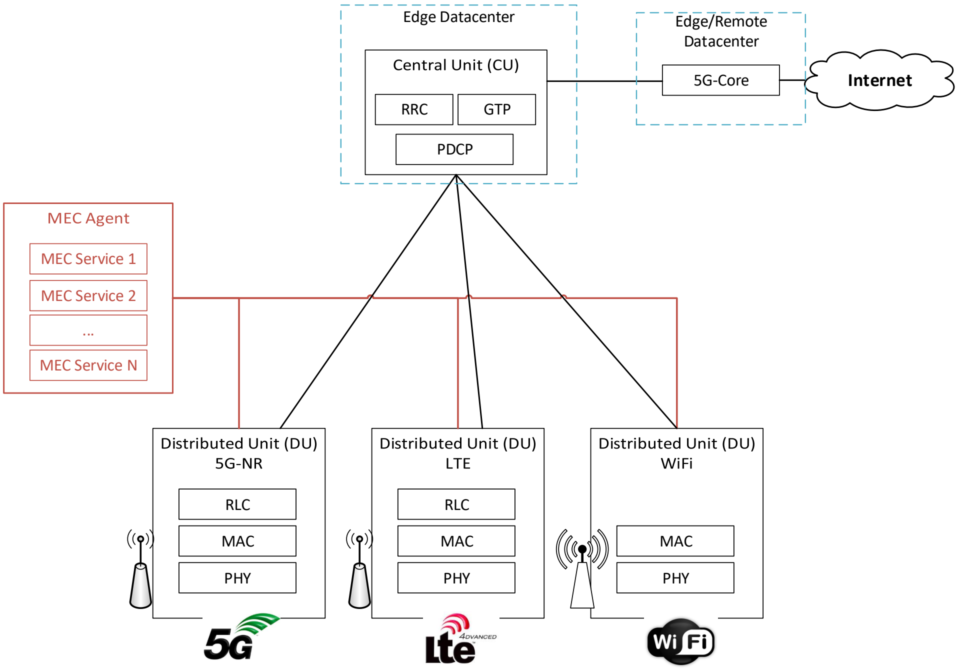


FIGURE 5.2: Proposed network architecture with MEC services being placed on the fronthaul interface. The sketched interfaces with red color are part of the contributions of this work.

Ethernet-based encapsulation, they can be easily handled by services introduced in the fronthaul interface.

Leveraging these works, in [66], a prototype implementation was introduced to offer MEC services closer to the DU side of the network. Some indicative results showed that even for 10MHz channel bandwidth in LTE implementations of the DU, the UE to MEC service latency can drop below 10ms, sufficient to serve several 5G applications in terms of latency [38]. Similar works on the development of similar MEC functionality in such experimental setups include [43] and [62]; in the former, the authors employ SDN based assisted control of GTP packets inside the Core Network, and in the latter the authors implement the "bump-in-the-wire" method to intercept packets on the backhaul interface of an LTE eNodeB. The benefit of the first solution is that it allows for flexible scaling of the services and core network services that are hosted. Nevertheless, the MEC services are still collocated with the Core Network. On the other side, in [62] the authors place the MEC services between the Core Network and the eNodeB. This allows us to bring the service even further to the network Edge, however, the solution relies only on application space management of the GTP tunnels that raises several performance issues, such as limited UE-to-service throughput performance and higher latency due to the extra overhead of processing tunnel encapsulated traffic.

All the previous works focus on the system design of MEC systems. Heterogeneity at the access network when MEC services are present is addressed in [27], where the authors propose a MEC aware algorithm for user association in heterogeneous

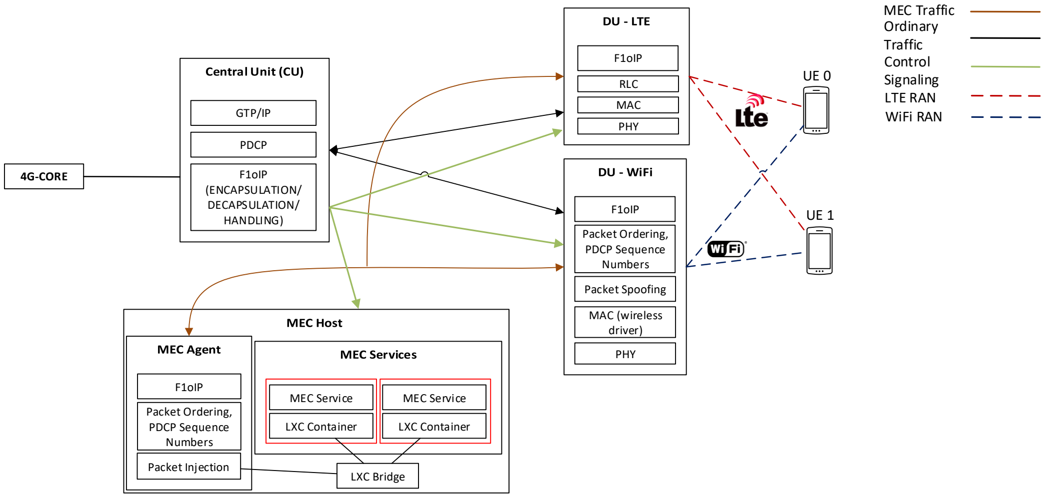


FIGURE 5.3: Proposed network architecture with MEC services being placed on the fronthaul interface, integrating the F1oIP [65] enhancements

dense networks. By appropriately selecting the radio technology at the UE side, the UE-to-service latency (only for the Uplink) is reduced up to 60%. Similar to this work, authors in [113] propose an algorithm for selecting the radio network used to offload cellular traffic when MEC services are provided. The authors focus on minimizing the aggregate energy consumption of the network's UEs.

In this work, we present a prototype that allows placing MEC functions on the fronthaul interface of heterogeneous disaggregated base stations, as shown in Figure 5.2, and use it to illustrate the performance benefits from placing the MEC applications close to the DUs of the network. The upper part of the base stations (CU) is executed at the network Edge, controlling multiple distributed units of different technology. UEs can access the network using multiple technologies (e.g. LTE and WiFi) and send their traffic through either the Core Network (CN) on the path DU-CU-CN or access services located close to the DU on the path DU-MEC-Service. The prototype is evaluated using different metrics and with dynamic adaptive video streams. In the next section, we provide an overview of the system architecture and underlying functionality developed in order to support such operation.

5.3 MEC System Architecture

The system architecture has been designed in a manner to allow the communication of three different entities: 1) a CU side, able to manage multiple heterogeneous DUs, 2) a DU, orchestrating the communication with the CU and the MEC services, supporting different technologies for the wireless access, and 3) a MEC Agent, communicating with both the CU for the exchange of control plane information and the

DU for receiving and transmitting data to the wireless network. For the implementation, we employ the OpenAirInterface [83] platform that is providing a software-based implementation of the LTE networking stack.

5.3.1 Disaggregated base station communication

According to the 5G-NR specifications, the disaggregated functionality of the 5G base stations shall address several different technologies. To this aim, the proposed standardized split option by 3GPP resides in the high layer 2 of the OSI stack, between PDCP and RLC layers. As this split option has slack requirements for the fronthaul link, it is an excellent candidate for accommodating multiple technologies, even non-3GPP compliant, like WiFi. In fact, the legacy LTE protocol is using the PDCP layer as the convergence layer for integrating WiFi in the RAN [3]. In the overall system communication between CUs and DUs, the relationship is 1:n, meaning that multiple DUs can be connected to a single CU. From the DU's perspective, this relationship is 1:1, so that each DU is associated only with a single CU.

In [65], we developed the F1oIP protocol as a communication mechanism between the CU and DUs of the system. The software is handling the Service Access Points (SAP) between the PDCP and RLC layers, by overriding the Service Access Point (SAP) functions between the layers: these are the *pdcp_rlc_data_request* for the Downlink (DL) traffic, and the *rlc_pdcp_data_indication* for the Uplink (UL) case. Instead of these, we introduce a communication mechanism based on asynchronous sockets between the two layers. Such a mechanism allows us to integrate other technologies by using an IP interface at the PDCP side. Under a monolithic setup, using the SAP interfaces, scheduling information is exchanged between the two layers intended for mapping the traffic to the logical, transport and subsequently physical channels of the network. The F1oIP implementation is piggy-backing this information in order to make such a transmission possible. The payload of these packets is a PDCP encapsulated packet, bearing a 2 byte long PDCP header. For the case that the CU is managing a non-3GPP DU (e.g. a WiFi DU), the same information is transmitted but stripped off before injecting the traffic to the wireless network. This information is further used in order to orchestrate the proper operation of the UL, by forming packets piggy-backing the information expected at the CU side.

For the UL case, the reverse process takes place before transmitting the packet to the CU. This means that the DU software is assigning new PDCP headers, based on the information exchanged between the CU/DU, and adds the respective information on the header in order for the packet to be handled at the CU side. For the WiFi integration case, the WiFi DU software generates new PDCP numbers, based on the traffic flow, generates the PDCP header and piggy-backs the information on the packets sent to the CU.

5.3.2 DU-MEC communication

As our target is to incorporate the services over the fronthaul interface, we need the appropriate interface to be developed between the DUs and the MEC server hosting the provided services. In [66], we developed a similar protocol for the DU to MEC communication, by introducing a *MEC Agent* component. This agent is able to generate and exchange the appropriate messages destined to the DUs of the network and receive and deliver the respective payload destined for the MEC services. The message exchange between the MEC agent and the DUs is taking place in a similar manner as with the CU-DU; when a DU has data that will be transmitted to the MEC service, it generates and transmits a *mec_data_request* message. This message is then handled by the MEC agent and delivered to the service. Similarly, for the service to UE path, the MEC agent generates and transmits a *mec_data_indication* message for the DU that the client is associated with. The DU information is discovered dynamically, based on the DU address that the agent last received a message from each UE. Figure 5.3 shows the overall architecture for the communication between the higher MAC layers of the DUs and the MEC agent service.

An important aspect here to consider is the ciphering process taking place at the PDCP layer of the network. As user plane data passes through the PDCP entity of the base station or the UE, it is ciphered according to the EPS Encryption Algorithm (EEA) chosen. Typically, there are four different variations of algorithms that are used [95]: 1) EEA0 - Null Ciphering Algorithm, 2) EEA1 - SNOW 3G, 3) EEA2 - AES and 4) EEA3 - ZUC. Therefore, the data that is exchanged below the PDCP point are ciphered, and the proper decryption mechanisms need to take place in order to retrieve the user data. For this purpose, we introduce a control packet that is broadcasted from the PDCP entity to all the DUs and MEC agents that are operating in the system, in order to ensure the deciphering process. Similar to the encryption case, we introduce extra signaling across the different entities of the network (CU, DUs, MEC Agents) in order to accommodate multiple clients over heterogeneous DUs. This includes mapping a cellular network UE with its respective non-3GPP interface and the manner that the different DUs identify it. We further detail how this is achieved in the following subsection.

5.3.3 Support for Multiple multi-homed UEs

Our target setup is considering multi-homed UEs, being served through more than one technology at the same time. Nevertheless, cellular base stations are merely seen as a Layer 2 device from the UE side: the connection established between the UEs is with the core network in the context of a PDN within the same IP address space. Therefore, as the base stations are disaggregated at the PDCP layer, and new DUs and MEC Agents are introduced to the system, functionality has to be developed to support IP based exchange of data between the UEs and the services on the fronthaul. When the cellular UEs are attached to the network, they are addressed by

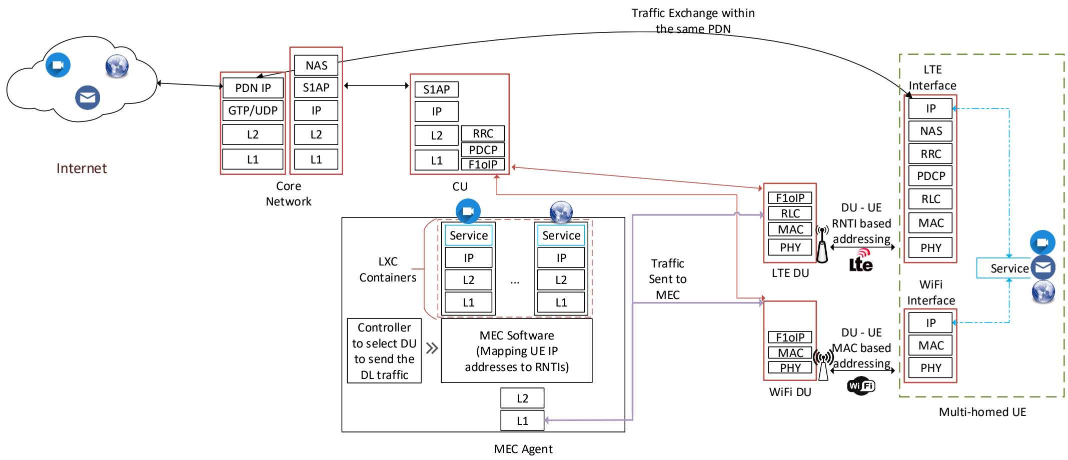


FIGURE 5.4: RNTI to IP mapping is realized at the WiFi DUs and MEC Agents in order to properly address the clients of the network; through a controller interface we are able to select the technology through which each user will be served for the downlink channel from the MEC service

the base stations using a Radio Network Temporary Identifier (RNTI). This RNTI is used by the base station at different layers of operation in order to forward the user plane data to the UEs, mapping them to the different logical and transport channels, etc. Contrary to this, for the WiFi case and the MEC services, the UEs are identified using IP addresses. This allows them to be addressed and request services from the MEC agent-based merely on the IP configuration of both sides (service and UE).

In order to cope with this problem, we introduce new signaling to the network as follows: whenever a new client is registered with the cellular DU, and the RRC process at the CU allocates a new RNTI to this UE, this information is broadcasted to all the different DUs and MEC agents in the system, in the form of an *rnti_inform* message. This message contains information about the RNTI of the client, an identifier (which client of the multi-RAT network it is), and through which DUs it can be served. This information allows us to create a mapping between the RNTI and the IP address that will be allocated by the Core Network to the UE, and be able to distinguish between them during the operation of the network. The RNTI information is actually being piggy-backed by both the DUs and MEC Agents of the network when sending data to the CU or the cellular DU respectively. This mapping allows us to efficiently map multiple services to multiple UEs being served through multiple technologies (see Figure 5.4). Through this functionality, we are able to create a controller process, running on the MEC Agents of the network, through which we select the DU that will be used to transfer the downlink traffic for each UE. This means that the MEC Agent will send the service traffic only to the DU that we select, and this DU will be used for forwarding the traffic to the end-user. In case that there is no such selection, by default, the agent sends traffic to the DU through which initially the UE transmitted traffic. This functionality for exchanging the RNTI information and selecting each DU that will be used makes use of the control channel introduced between the different entities of the network, as shown in Figure 5.5.

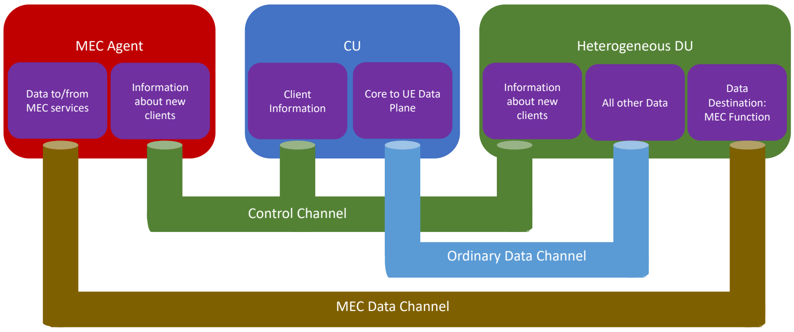


FIGURE 5.5: Intercommunication architecture of all the involved entities over the fronthaul interface

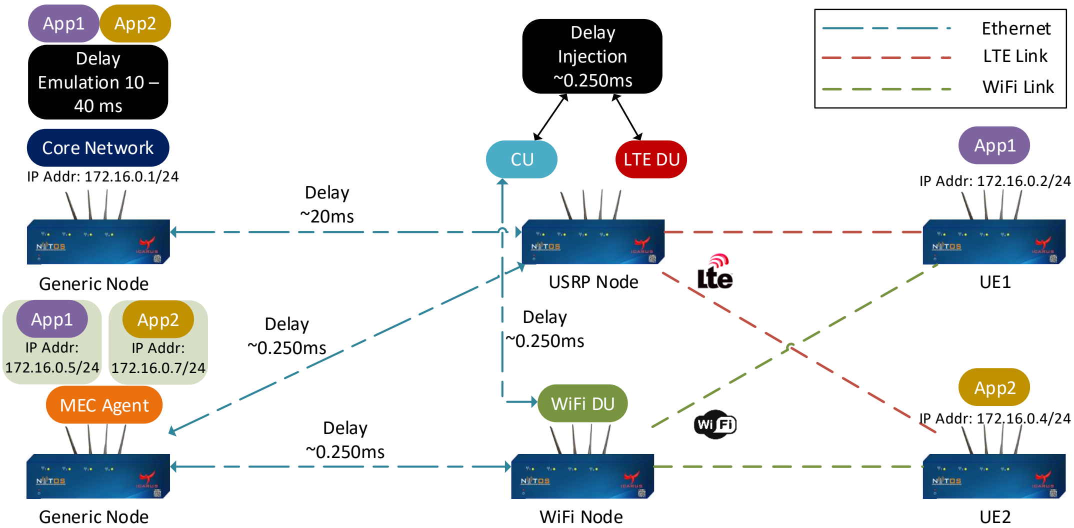


FIGURE 5.6: Experimental topology for the evaluation of the MEC scheme in the NITOS testbed

5.3.4 Support for multiple MEC services

One of the key components in our scheme is the MEC Agent, as it orchestrates the communication of the services running on it with the DUs of the system. Whenever the agent receives MEC intended traffic, it decapsulates it and injects it to the MEC service. We select to run the MEC services as containerized services through Linux Containers (LXC), as they can be instantiated on the fly, whenever an end-user requests different services from the MEC platform. Employing LXC containers has multiple benefits; it allows each new service to be addressed with a new container, with a new network IP address, that can be easily migrated if needed to another edge host, like for example in the case of a rapidly moving mobile UE (V2X case). As the LXC service places all the containers with different IP addresses under a bridge interface on the edge host, the MEC agent has to inject the traffic to the bridge, destined to the MAC address of the container implementing the MEC service. Through the mapping described in the previous section, between the RNTIs and the IP addresses of the services and the UEs of the network, multiple UEs can make use of the same service, even when they are getting connectivity through different access technologies.

5.4 Experimental Setup

In this section, we showcase our experimental setup and methodology for experimentation. The described functionality has been developed over the OpenAirInterface platform (OAI) [83], that provides an open-source software implementation of the cellular base station stack and can be executed over commodity hardware with the appropriate Software Defined Radio front-ends. We conduct the experiments over the NITOS testbed [67]. NITOS is a heterogeneous testbed located in the premises of University of Thessaly, in Greece, offering a rich remotely accessible experimentation environment with resources spanning from commercial LTE, to WiFi and Software Defined Radio platforms that suit our experimentation needs.

TABLE 5.1: Testbed Equipment parameters for experimenting with MEC

Network Parameters	Values
LTE mode	FDD Band 7
LTE Frequency	2680 MHz (DL)
Antenna Mode	SISO
No RBs	25 & 50 (5/10 MHz)
UE	Cat. 4 LTE, Huawei E3272
WiFi Technology	802.11n MIMO 3x3
WiFi Channel BW	40 MHz
WiFi card	Atheros 9380
Back-/Front-haul RTT	$\sim$ 0,450 ms
Back-/Front-haul capacity	1Gbps Ethernet
Ethernet MTU size	1400 bytes
Video Client	VLC v. 2.1.0 with MPEG-DASH
Video File	1080p AVC1 transcoded
Video Duration	60 secs in samples of 1 sec

We focus on the LTE implementation of OAI, as it provides the functionality for the high layer splits compared to the recent 5G-NR release. We employ an altered version of the WiFi DU module developed in [65] in order to set up a separate communication channel between each DU and the MEC Agent, and a control channel between the CU and all the DUs that transmits the RNTI related information for UE to service mappings (see Figure 5.5). This channel and the F1oIP channels for the CU/DU communications are selected to be TCP over Ethernet, as our former experiments denote that there is no notable performance degradation compared to UDP or even the vanilla OAI setup. Google's Protocol Buffers Library [88] has been used for the generation and handling of the messages over the involved entities.

The MEC services are loaded on a node using the LXC framework for providing containerized MEC services. For every packet that the MEC Agent is receiving, it is deserialized from the F1oIP protocol and injected to the *libvirt* [40] bridge for addressing the containers providing the services. Service differentiation is reflected to different IP addresses; hence, a video service is running on a container using a different IP address than a simple traffic generator application. Both of these addresses are within the same address space that the UE is using to communicate with the Core Network.

TABLE 5.2: RTT Results (in milliseconds) for LTE and WiFi access to the service when tuning backhaul delay up to 5ms

	LTE to FH	**WiFi to FH**	**LTE to EPC**	**WiFi to EPC**	**LTE to EPC (5ms)**	**WiFi to EPC (5ms)**
Avg. RTT	19.7	4.78	32.32	5.26	36.66	9.09
Min. RTT	15.1	4.39	26	4.59	29	8.6
Max. RTT	24.7	5.12	43.4	6.64	48.9	9.73

TABLE 5.3: RTT Results (in milliseconds) for LTE and WiFi access to the service when tuning backhaul delay up to 20ms

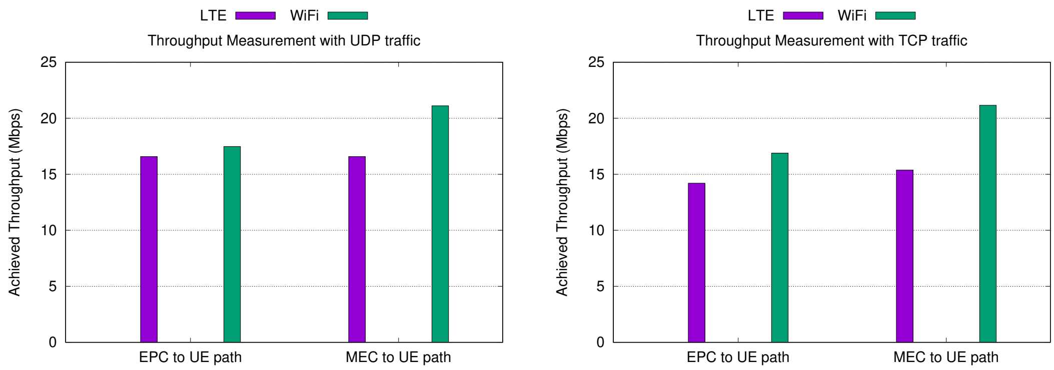
	LTE to FH	**WiFi to FH**	**LTE to EPC (10ms)**	**WiFi to EPC (10ms)**	**LTE to EPC (20ms)**	**WiFi to EPC (20ms)**
Avg. RTT	19.7	4.78	41.58	15.19	51.8	25.14
Min. RTT	15.1	4.39	32.9	14.5	40.8	24.4
Max. RTT	24.7	5.12	61.9	17	59.9	25.7

We employ different services in order to measure the performance of the understudy scheme. For testing video services, we employ an MPEG-DASH server [102], able to stream videos of up to 1080p resolution, for video segments of 1 sec. This means that the client running on the UE side can request a video segment for the next second that will be played from a selection of different transcodings. The server is running over an Apache2 web service, in the MEC containers and the Core Network for comparing their performance. Each DASH client initially requests a Media Presentation Description (MPD) file from the server. According to the descriptions of the available video segments and the video requesting algorithm running on the application, the video is downloaded to the client. We use VLC as the end-user application, based on the policies that are described in [36]. The policy that we use for streaming the video is the following: for each video segment, VLC estimates the channel's download rate. For the next segment to be downloaded, it will request the video with coding rate equal to the download rate. In the case that it does not exist (since the video coding rate might be significantly lower than the actual channel rate), it will request the next lower representation available. For the cases that the video buffer is less than 30% occupied, the client will request the lowest video representation available. Using this policy we want to measure the quality that each client of the network is served, based on their own observings of the network. We expect that when the video is streamed from the MEC service, the client shall be able to more quickly converge to the best video representation available.

The topology for our experiments is given in Figure 5.6. The current version of F1oIP is only allowing the data plane split between the CU and the LTE DU. Therefore, we emulate the disaggregated behavior by injecting delay between the network interfaces that are used for this communication between the CU and DU, equal to ∼0,250ms. The delay injection is done with the *netem* application and is in the range of the mean delay that we measure over the fronthaul interface between two nodes of the testbed. Table 5.1 is summarizing all of our experimentation parameters with respect to the wireless network configuration.

5.5 System Evaluation

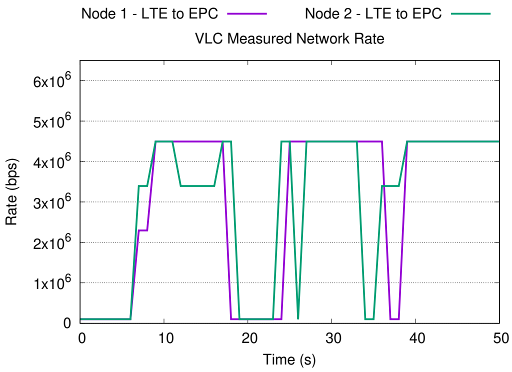
In this section, we present our experimental findings from evaluating the proposed scheme. We focus on two different performance indicators: 1) the latency time for reaching the MEC service vs a traditional deployment of the service on the Core Network (or beyond), and 2) the video streaming quality for multiple users when using either the MEC or the EPC video server. We organize the experiments in the following way: 1) Initial benchmarking experiments, for measuring the latency for reaching the Edge or core network service through different technologies, 2) experiments illustrating video performance when reaching the service through different technologies, and 3) experimentation for the placement of the video service (Edge vs Core) depending on the access technology. For all our cases, we use two multihomed UEs connected to two DUs (one LTE and one WiFi) and measure on the path between the UE and the service. The experiments are limited to two UEs, as the version for software base stations that we use (OpenAirInterface) behaves in an unstable manner for more.



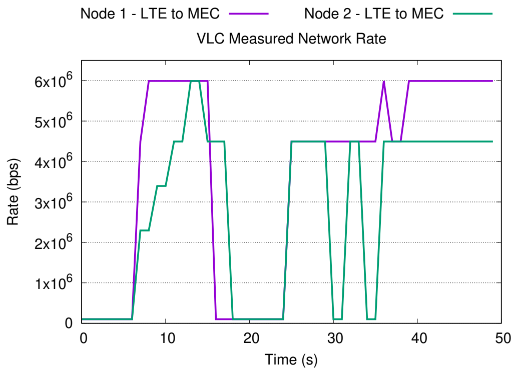
(A) UDP Throughput

(B) TCP Throughput

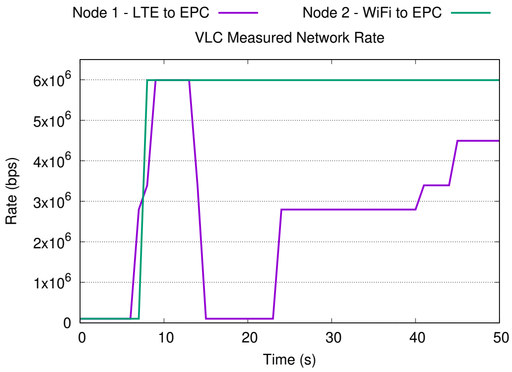
FIGURE 5.7: Service to UE maximum throughput



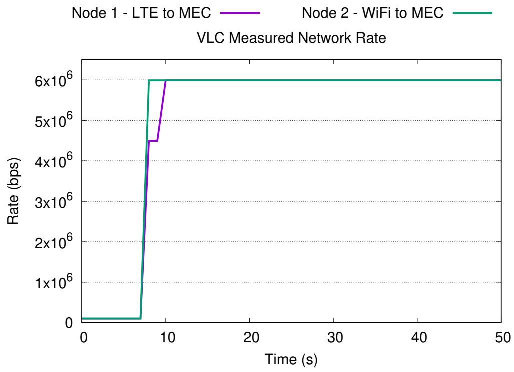
(A) VLC Rate - 2 LTE UEs requesting at the EPC



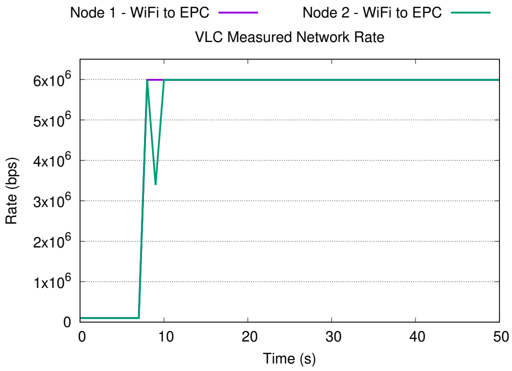
(B) VLC Rate - 2 LTE UEs requesting at the MEC



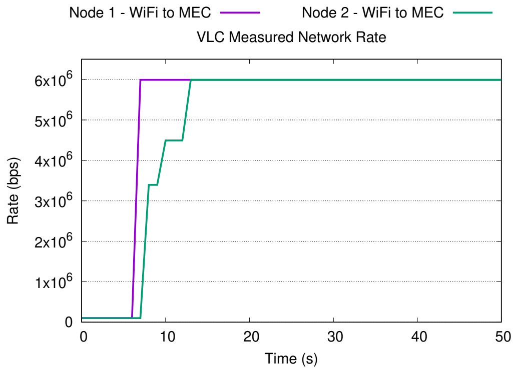
(C) VLC Rate - 1 LTE and 1 WiFi UEs requesting at the EPC



(D) VLC Rate - 1 LTE and 1 WiFi UEs requesting at the MEC



(E) VLC Rate - 2 WiFi UEs requesting at the EPC

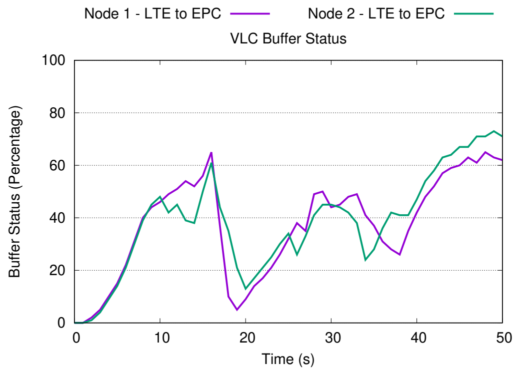


(F) VLC Rate - 2 WiFi UEs requesting at the MEC

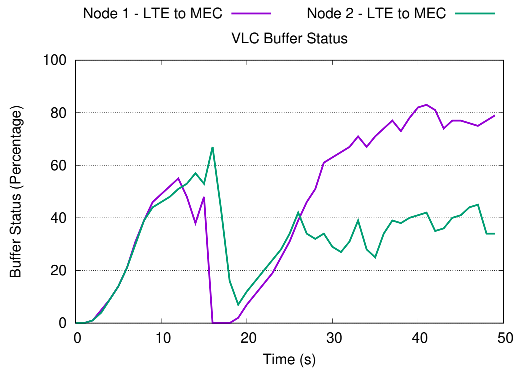
FIGURE 5.8: VLC rates for different access technologies

5.5.1 Latency and Throughput measurement

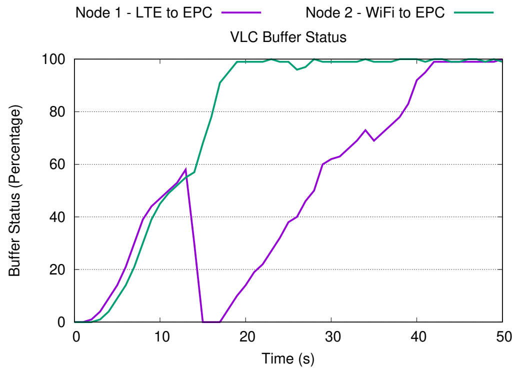
We compare the latency time for both access technologies between the UE and the service using two different deployments for the service: one being on the fronthaul, with approx 0.250ms delay between the DU and the MEC agent, and one being on the core network. As in typical deployments, the core network is not located so close to the edge, we measure the link for the cases of no latency and for tuning the latency for accessing the service. Thus we get an emulated behavior that the services are deployed at distant servers for typical values of latency (e.g. San Francisco to New York is approx 20ms).



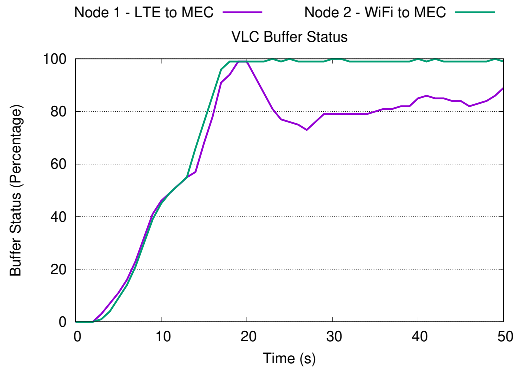
(A) Buffer Status - 2 LTE UEs requesting at the EPC



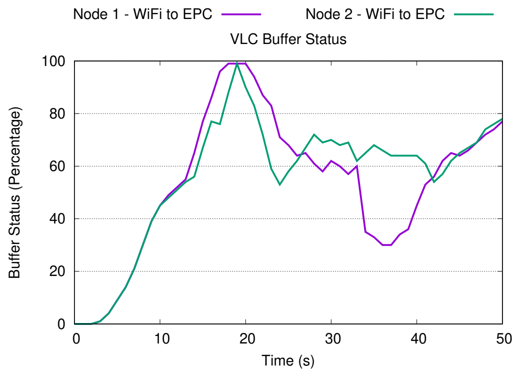
(B) Buffer Status - 2 LTE UEs requesting at the MEC



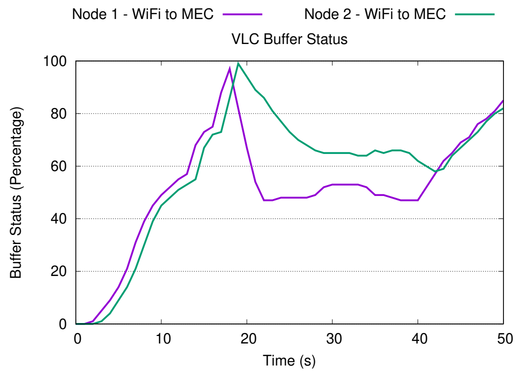
(C) Buffer Status - 1 LTE and 1 WiFi UEs requesting at the EPC



(D) Buffer Status - 1 LTE and 1 WiFi UEs requesting at the MEC



(E) Buffer Status - 2 WiFi UEs requesting at the EPC

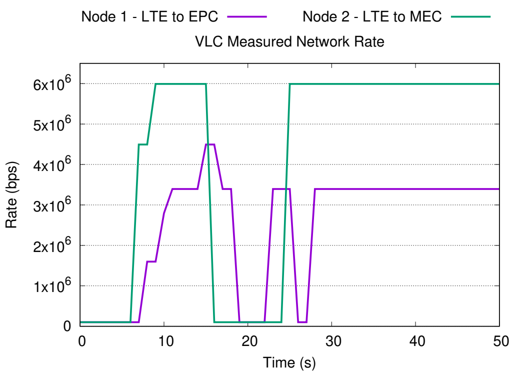


(F) Buffer Status - 2 WiFi UEs requesting at the MEC

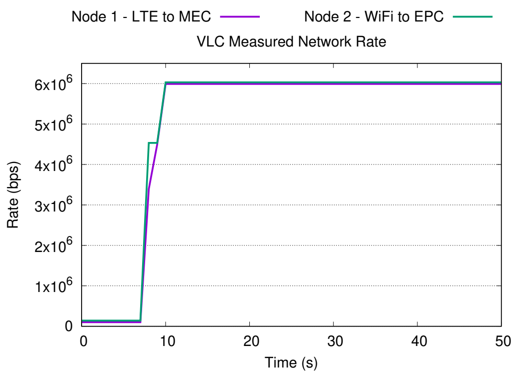
FIGURE 5.9: Buffer Status for different access technologies

Table 5.3 shows indicative RTT times for accessing a service located as a container on the MEC agent or the Core Network (EPC) when accessing the network through either the LTE DU or the WiFi DU. Assuming that latency is almost half of the RTT time, we see that for the cases of MEC access over LTE or WiFi, the latency is consistently less than 10ms, thus allowing several 5G applications to run according to [54]. As we run the experiments in a completely isolated from external interference environment, we see that WiFi outperforms the LTE for the cases of latency, even when tuning the delay on the link between the CU and the EPC.

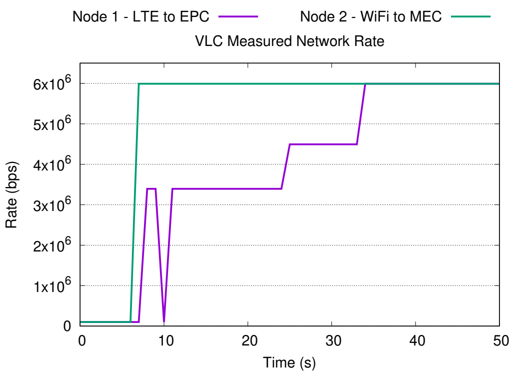
For the sake of completeness of experiments, and to better understand the results from the next subsections, we perform a throughput test for both technologies from the UE to the MEC and the EPC services. Figure 5.7 shows our results for the downlink channel, when saturating the link with UDP traffic (Figure 5.7a) or when using TCP streams (Figure 5.7b). For all the remaining experiments, we use a 5MHz LTE channel as it provides us with more stable links. We see that for the EPC to UE paths, for the UDP case both LTE and WiFi technologies get approx. 16.5 Mbps links, whereas for the MEC to UE path we get similar performance for the LTE access and slightly enhanced for the WiFi access (up to 21Mbps). When using TCP, which is also the case for the experiments illustrated in the next sections, we see that for the EPC to UE path LTE gets slightly less than 15Mbps, while WiFi access achieves up to 16.5Mbps. For the MEC to UE path, we see that the path involving an LTE link gets 15.5Mbps whereas the WiFi path gets 21Mbps. These results provide us with some insights on what to expect in the video streaming results: accessing the MEC service from an LTE connection should perform better than the same service being placed in the EPC. Similarly for the WiFi case, the same assumption holds.



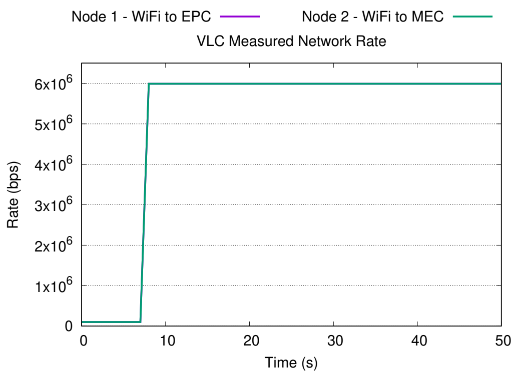
(A) VLC Rate - 2 LTE UEs requesting at the EPC and MEC servers



(B) VLC Rate - 1 LTE UE requesting at the MEC and 1 WiFi UE at the EPC



(C) VLC Rate - 1 WiFi UE requesting at the MEC and 1 LTE UE at the EPC



(D) VLC Rate - 2 WiFi UEs requesting at the EPC and MEC servers

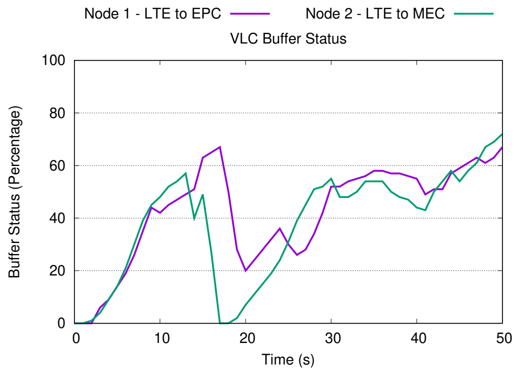
FIGURE 5.10: VLC rates for different placement of the service and different technologies at the UEs

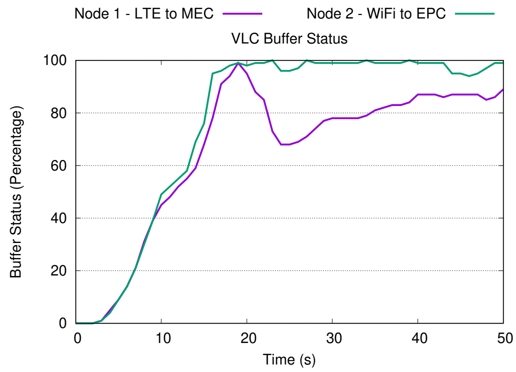
5.5.2 Video measurement

For the second part of the evaluation, we test the network with two UEs, connected through either LTE or WiFi and request the video from a server located either at the EPC (similar to the case of Figure 5.1c) or the MEC server (case of Figure 5.1a). We plot the requested video rate of the application based on its assumption of the underlying wireless channel, and the current buffer status for the video depicted at the end-user. We remind here that if the buffer status is less than 30%, the minimum representation possible is requested, and thus the reported rates from the application are very low. The plotted video rate is representing the application's choice for video transcoding, based on its perspective on the wireless channel.

Figure 5.8 shows the results on the selected video rate, and Figure 5.9 the results on the buffer status of each UE. For the cases that both users use the LTE connection, the selected video coding rates for the application are get barely over 4.5 Mbps. Also, as both users share the same channel, they struggle to get the best video segments that are available and hence their buffer status is kept lower than 50% for most of the experiment time. When requesting the video from the MEC server over LTE (Figure 5.8b), one of the two UEs manages to get video rate coded at 6Mbps, whereas the second one is bounded at 4.5Mbps, similar to the EPC case.

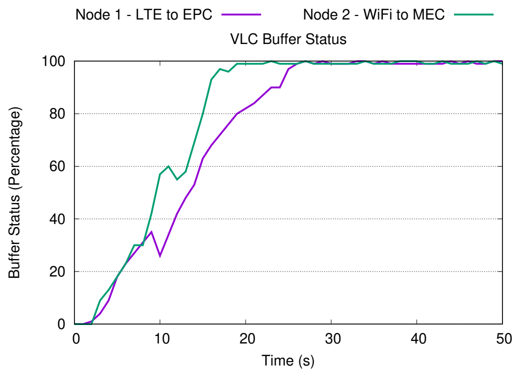
When we use different technologies (one user to LTE, one to WiFi) to request data from the EPC server (Figure 5.8c), both clients get video transcodings at 6Mbps, until the LTE UE's buffer is emptied. Then it gradually starts getting better video segments up to 4.5 Mbps. On the other side, the WiFi client quickly converges to getting the best video quality available. When using the same setup to get video from the MEC service, we see that both clients quickly converge to receiving the best available video quality (Figure 5.8d), and their buffers are kept full for most of the experiment time (Figure 5.9d). For the cases that both users use only the WiFi technology to request the video content from the EPC (Figure 5.8e) or the MEC (Figure 5.8f), we see that both clients quickly converge to the best available video quality (6 Mbps) for both cases. Similarly their buffers (Figures 5.9e and 5.9f) are constantly kept above the threshold of 30% that we have set for requesting the best video available, based on the UEs perception of the channel quality. This is happening due to the slightly better throughput that the UEs get when using the WiFi channel (as shown in Figure 5.7b) and the lower latency times that the clients have. We also observe that for the scenario of both UEs using the WiFi channel, due to channel contention between the UEs, the second node is gradually enhancing the requested video quality. From these results, we conclude that the technology used to request the video plays a key role in the overall experience of the user, whereas the services that are placed on the MEC agent and therefore are closer to the UE outperform the cases of remote testbed placement.

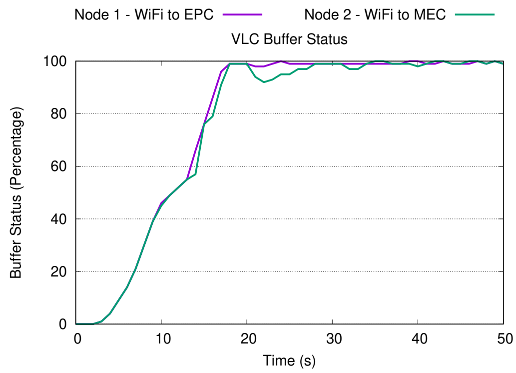




(A) Buffer Status - 2 LTE UEs requesting at the EPC and MEC servers

(B) Buffer Status - 1 LTE UE requesting at the MEC and 1 WiFi UE at the EPC





(C) Buffer Status - 1 WiFi UE requesting at the MEC and 1 LTE UE at the EPC

(D) Buffer Status - 2 WiFi UEs requesting at the EPC and MEC servers

FIGURE 5.11: VLC buffer for different placement of the service and different technologies at the UE

5.5.3 Different placement of services

As a final set of experiments, we demonstrate results when the placement of the service is different for the two UEs, i.e. one requesting at the MEC server and the second one at the EPC, concurrently. This scenario is showing results of how the application provider could discriminate between different subscription plans of clients, allowing certain subscribers to access the same service located at the edge datacenter contrary to the rest. As our prior experiments denoted that the access technology plays a significant role in the performance that the UE application gets, we experiment with both the available technologies and for all the different placements of the service.

Figure 5.10 shows the plotted VLC rates for the different placements of the video server and the different access technologies. Figure 5.11 shows the respective buffer status for each setup. For the case when both of the clients use the LTE channel and request video from the MEC and the EPC servers concurrently, we observe that the UE requesting from the MEC service is able to get better video quality (Figure 5.10a). The node requesting to the MEC service very quickly converges to the highest rate (6 Mbps) whereas the node requesting at the EPC barely manages to get just over 3 Mbps. We see that at approx. 15 seconds of experimentation, both UEs drop their requested rate to the minimum possible, due to their buffer that is emptied. However, when they start requesting video samples again, the MEC requesting UE is getting the highest possible performance.

When we introduce one WiFi UE, we see that regardless of where the service is deployed (MEC/EPC) the WiFi UE is able to very quickly converge to the highest video representation available (Figures 5.10b and 5.10c). We see that for the same cases the LTE UE is able to request the highest representation available. However, in the scenario that the LTE UE is requesting the data to the MEC service, the rate is reached very rapidly in the experiment (less than 10 seconds) whereas for the case that the LTE UE is requesting at the EPC service, it takes just over 30 seconds to request the highest video representation available. This is happening entirely due to the location of the service closer to the edge, as we see that both buffers quickly fill with over 70%. For the experiment when both UEs use only WiFi to access either at the MEC or the EPC server, we observe that both of them quickly fill their buffer and request very rapidly the best video representation available. This fact is happening due to the slightly better capacity of the WiFi medium for our experiments, as shown earlier in this chapter.

5.6 Discussion

The obtained results denote that the closer that each service is placed to the access network, the better the performance is related to the Quality of Experience (QoE) of each user. This is clearly illustrated in the better video quality requested from the LTE clients when using the MEC server, or for the WiFi clients, as the WiFi network has slightly better capacity for our experimental settings. The closer the service is placed to the DU, the lower are the response times for our tested applications, and thus the better the convergence times for requesting better video quality compared to a distant server.

The placement of services closer to the DUs, allows the service providers to create different subscription models for the served UEs. For example, consider a paid subscription plan for a service that ensures always the least possible latency times compared to a standard plan. This creates the opportunities for the operators to examine the problem in different manners, such as which technology will be used in a per-packet basis to serve a specific UE, where the service shall be placed subject to UE mobility, DU load, spectrum utilization, and maximum serviced traffic from the server based on the available computational resources. Service migration shall also be considered, with a seamless approach, allowing the UEs to experience no service breaks while the service is transferred between hosts, and also a new path is established to the MEC service. For the latter, methods such as DNS cache poisoning can be exploited in order to create a localized illusion that the service is still on the same host. For the former, virtualization of services and related work provides several insights on this [55].

Important aspects to consider for the placement of the services are also the conditions of the wireless network. A highly utilized DU, or a WiFi DU operating in an ultra-dense setup with several other DUs operating within the same band may create additional delays for service access time. Moreover, using unlicensed bands poses several performance-related issues, as the medium conditions cannot be accurately predicted. For these cases, and taking into consideration the different traffic patterns of UEs in an area, a machine-learning approach for determining where the service shall be placed can play a key role in the overall efficiency of the network and the provided services. By monitoring the mobility patterns of the UEs, and the requested content at a central location (e.g. the Core Network of the cellular infrastructure), we might efficiently determine where the service can be placed (e.g. close to the next target DU) or the radio access technology through which each client can be served, in order to ensure the best possible QoE.

5.7 Chapter Conclusion

In this chapter, we propose and experimentally evaluate moving services closer to the network edge, when employing a disaggregated heterogeneous base station setup (Cloud-RAN). Based on our former works, we extend our prototypes and provide a fully-fledged solution for placing the MEC services on the fronthaul interface of multi-technology base stations, complying with the Cloud-RAN concepts. The developed functionality is demonstrating the performance gains in terms of latency and overall QoE that mobile multi-homed UEs can gain when the service is deployed even closer to the network edge. The experiments denote that the selection of the technology for serving each UE can widely affect the QoE. However, as our experiments only use a single technology for the entire traffic stream, possibly better results could be received if traffic is balanced between multiple DUs.

In the future, we foresee extending the scheme in order to include a machine-learning approach on the manner that we select where the service is placed. Through monitoring the traffic patterns of each UE at the core network, and taking into consideration the wireless conditions at each DU and reported channel qualities from the clients, we plan to develop an algorithmic scheme in order to define where each service shall be placed, in order to ensure each UE's QoE. This service instantiation can take place in a completely seamless to the UE manner, by exploiting the migration features of the containers hosting the network servers [100] and using a DNS spoofing method at each DU in order to redirect the requests of each user to the new location of the service. A similar approach can be further applied in order to define the balance between the DUs for transferring the traffic to each UE of the network.

Chapter 6

Service Orchestration over Wireless Network Slices

Contents

6.1 Chapter Introduction

Network Functions Virtualization (NFV) aims at adding up to the flexibility, overall sustainability and management of the infrastructure that network operators use to provide services. Through the introduction of fully softwarized network architectures, the deployment of Network Functions (NFs) may take place with a single click fashion, offering easy remote updates, contrary to installing traditional hardware equipment with prior planning. The key features of NFV include reduced CAPEX and OPEX costs, optimized reconfiguration of the provided services, easy integration of new services and operational efficiency due to the concepts of network slicing and multi-tenancy (i.e. multiple operators using the same physical infrastructure). Due to the above reasons, NFV is considered to be one of the major enablers for the 5G technology [112].

In order for the orchestration process to take place, the underlying compute infrastructure needs to be prepared in terms of networking and resource allocation (e.g. CPU and Memory available from the host machine). Management and Orchestration (MANO) of the hardware resources is not a trivial task, as the interplay of the heterogeneous compute equipment, providing different APIs for the configuration of different components, needs to be efficiently defined. To this aim, the NFV-MANO [28] working group was formed by ETSI with the purpose of defining a generic architecture for the management and orchestration of virtualized resources. The hardware resources (compute, storage, network) are abstracted through the framework, whereas focus is placed on the efficient interconnection of the orchestrated components.

Nevertheless, NFV-MANO has been developed considering mainly datacenter resources, with the networking being programmed through the SDN concept, and services being deployed using either virtual machines or light-weight containers. The advent of fully softwarized architectures for the RAN creates fertile ground for the re-consideration of services deployed in the network, moving beyond just application servers towards full stack networking systems. This fact also expands for wireless networks as well, with the ability to setup fully softwarized 5G base stations [83] with an appropriate radio front-end device (Software Defined Radio - SDR). Considering that such applications usually assume distributed compute infrastructure, like for the case of deploying a base station along with a Core Network, the compute infrastructure needs to extend to generic networking devices, including heterogeneous equipment for the wireless operation. These technologies are not currently addressed by any orchestrator or through SDN production grade software.

In this chapter, we present our contributions towards integrating wireless resources in an NFV-MANO compliant orchestrator. All of our contributions are experimentally driven, applied to an open wireless testbed, and are packed in a fully-fledged solution easily adaptable by similar testbeds. The main contributions of our work are the following:

- To combine wireless resources in the NFV-based orchestration of services over real resources.
- To support the deployment of VNFs which use wireless links, through LTE, WiFi and mmWave technologies transparently, without altering the information model for the intercommunication of the different entities in the NFV-MANO architecture.

- To provide and apply entirely isolated slices of the hardware infrastructure for the hosted VNFs, by exploiting solutions that can be applied on off-the-shelf equipment.
- To deploy and evaluate a novel use case over the testbed infrastructure, dealing with the deployment and operation of 5G networks.

As our orchestrator solution we employ the Open Source MANO (OSM) [30] platform, the most widely adopted framework for NFV-MANO. Our goal is to extend the orchestrator and underlying Virtual Infrastructure Manager (VIM) services in order to provide services running on top of different wireless networks deployed in the testbed. Traditionally, tools like OSM do not deal with network connections other than Ethernet, and make use of network interface virtualization enablers such as SR-IOV. Moreover, none of the widely available VIMs support configuration for the wireless network. As similar solutions for the virtualization of the wireless interfaces also exist, they can be used in order to provide slices of wireless connectivity to orchestrated services, even for off-the-shelf equipment. The resulting infrastructure is a powerful platform allowing the orchestration of experiments using virtual resources, that off-the-shelf supports portability from any other OSM-compliant site to the testbed.

The rest of the chapter is organized as follows: Section 6.2 is providing some former practices on the virtualization and orchestrator support for key wireless technologies. Section 6.3 is proving a brief description of the testbed's resources and former experimentation methodology. Section 6.4 is detailing our architecture for provisioning the testbed with the NFV-MANO paradigm. In section 6.5 we detail our scheme for enabling the slicing of the wireless platforms and section 6.6 details the integration of the orchestrator with the existing tools for provisioning the testbed to experimenters. In section 6.7 we benchmark the developments in our platform and illustrate a 5G related use case scenario, orchestrated through our contributions. Finally, in section 6.8 we discuss the experimentation potential of the platform and conclude.

6.2 Related Work

Related research focuses on either the development of new orchestrators, or the development of slicing mechanisms for hosting multi-tenant operations over the distributed infrastructure. Below we initially list some relevant literature on the orchestration of heterogeneous equipment, and subsequently we provide the state-of-the-art for wireless network virtualization.

Although there is a plethora of different orchestration software (e.g. OSM [30], ONAP [63], Hurtle [45], SONATA [26], Kubernetes [13] etc.), most of them fail to address wireless resources, or when they do so, they only focus on the SDN based integration of wireless network segments for inter-networking the compute nodes.

Nevertheless, apart from the ONAP case, all of them fail to address wireless resources, or when they do so, they only focus on the SDN based integration of wireless network segments for inter-networking the compute nodes. The Open Network Automation Platform (ONAP) on the other hand is a framework created by the merging of the ECOMP (Enhanced Control, Orchestration, Management & Policy) and Open-O (Open Orchestrator) projects, in order to bring the capabilities for designing, creating, orchestrating and handling of the full life-cycle management of Virtual Network Functions, Software Defined Networks, and the services that all of these things include. In this context, ONAP goes beyond the current NFV-MANO architecture, and integrates several functions dealing with the configuration of wireless networks as well. As a matter of fact, ONAP has a dedicated working group for the development of the appropriate functionality for deploying the Radio Access Network (RAN) of 5G networks as well [85]. Nevertheless, the complexity of ONAP and the amount of resources needed for running the core framework hinder its application beyond operator use cases [111]. The most widely adopted framework that complies with the NFV-MANO specifications is OSM [30], which we also adopt in this work. Similar attempts that comply with the NFV-MANO specifications include [74], which is built around the feature of providing multiple network slices to different tenants of the infrastructure. Similarly, in [60] the authors present an orchestration software that supports multi-tenancy. Authors in [34] and [33] address the multi-domain management and orchestration, with domains belonging to different entities or being geographically distributed. Similarly, the work of the EU 5GEx project [12] resulted in providing the specifications and prototype implementation of a multi-domain orchestrator.

However, all these works fail to address the wireless parts of the network, and services being deployed in need of wireless connectivity. An attempt to address this is presented in [106] where the authors use SDN controllers hierarchically organized in order to manage a converged optical and wireless infrastructure. In [96], the authors organize through SDN wireless small cells, by using an LTE connection as their control channel. The Maestro platform [23] provides an NFV orchestrator for wireless environments, aware of the features of the VNFs that it is going to deploy. Similarly, in [51] the authors build an orchestrator that can deploy software base stations based on the OpenAirInterface platform [83] and compatible radio front-ends (SDRs). In [69] we presented an initial approach for extending the OSM orchestrator in order to provide services with wireless connectivity. However, the information model for the communication between OSM and the VIM is altered, and thus it fails to provide portability of VNFs across different platforms. In this work, we extend the initial approach in order to introduce our contributions transparently to the existing orchestration services, and thus enable the portability of services through different orchestrators.

Since one of our goals is to provide slices of the wireless infrastructure, we present some relevant literature on this topic. There exist several methods for slicing the wireless networks, even with guarantees, depending on the technology being used. In [94], the authors list all the relevant methods for slicing wireless networks. These methods may rely on either introducing changes to the internal scheduler of each technology, or make use of a middleware software in order to control the transmission of information to each technology. An example of the latter is [46], where the authors use different traffic classes and allocate them to multiple radio access networks in order to prioritize transmissions of different slices over the overall infrastructure. Contrary to this, in [49] the authors allocate different resource blocks for the downlink channel of LTE networks allocated to different slices.

However, most of these works solely focus on simulations and do not apply the slicing mechanisms over real networks. Authors in [52] and [81] provide a solution for the OpenAirInterface implementation of the LTE stack, by exposing the MAC scheduler and allocating resources from the LTE wireless network to different UEs. Similarly, authors in [58] create a queuing mechanism before the MAC scheduler for prioritizing different user of the network. The authors use a real WiMAX testbed in order to evaluate their approach. Works in [15] and [14] use an external service in order to slice a WiMAX base station for different flows of traffic. Slicing is also expanding to WiFi networks as well (e.g. [93], [110]) where the medium is accessed opportunistically and guarantees for different slices are hard to meet. Authors in [61] and [53] develop a scheme based on the multi-VAP feature of hostapd [72] and ClickOS [73] in order to guarantee the air times of each slice.

In this work, we slice commercial off-the-shelf (COTS) infrastructure for LTE, WiFi and mmWave technologies, with the aim to provide guarantees for the deployed slices in terms of throughput for the LTE and mmWave case and in airtime for WiFi, allocated to different tenants of the infrastructure. For the case of the testbed, different tenants are different experimenters who need to access the resources in an entirely isolated manner. The work is integrated in the OpenVIM installation of the tested, appropriately enhanced in order to provide deployment of VNFs over wireless technologies without any changes in the information model between the orchestrator and the VIM infrastructure. All the developments are integrated in the NITOS wireless testbed [84], which provides access to host machines, mainly targeting experimentation and prototyping for wireless networks. In the following section we briefly present the testbed, and in the subsequent section how the experimental infrastructure is managed as our compute infrastructure from the VIM side.

6.3 NITOS Testbed Environment

The target facility used for the development and applications of our proposed extensions to the orchestration software is the NITOS testbed (`http://nitos.inf.uth.gr`), located in University of Thessaly, Greece. The testbed is providing in a 24/7 fashion remotely accessible resources, targeting at experimentally driven research in wireless networks. The testbed is providing access to over 100 physical nodes, equipped with key technologies:

- All the nodes are high-end PCs, equipped with Core-i7 processors and 8 GBs of RAM each, featuring at least two IEEE 802.11 a/b/g/n/ac cards, compatible with Open Source drivers (e.g. ath9/10k) used for WiFi related research.
- Two commercial off-the-shelf LTE access points are available for experimentation, along with the respective Core Network solution. Both femtocells and core network are programmable through the available testbed services [67]. About half of the nodes are equipped with LTE dongles, that allow the establishment of an operator-grade LTE network, using testbed specific SIM cards.
- Over 20 different SDR devices exist in the testbed, which are compatible RF front-ends for open source implementations of base stations (such as OpenAirInterface [83]).
- Six mmWave devices are installed in the testbed, reachable from all the nodes of the testbed, and supporting the creation of high-throughput wireless point-to-point links.
- All the nodes of the testbed are interconnected through different hardware Open-Flow switches, organized in a tree topology. Users can set their own controller to manage the flows of the nodes that they are using.

The testbed is organized in three different setups: An indoor RF-isolated, an outdoor setup prone to uncontrolled external interference and an office setup with mild interference settings. Resources can be mixed from the different locations in order to create a versatile experimentation environment. The nodes can be reserved through a portal service, for up to four hours per slot. Several tools are available for experimentation, supporting large scale deployments of ultra-dense heterogeneous wireless networking environments. The most notable of the tools is the cOntrol and Management Framework (OMF) [89] which supports receiving an experiment description written in a high-level language, and subsequently orchestrates the experiment over the physical machines of the testbed. OMF also manages the control plane part of the testbed for the experimenters, including the processes of loading new images on the compute nodes with PXE booting [35], turning them on/off etc. As the tool lies in the heart of the testbed operations, and allows exposing the testbed to external remote experimenters, our solution for orchestration takes this into consideration. We provide more details on this in section 6.6.

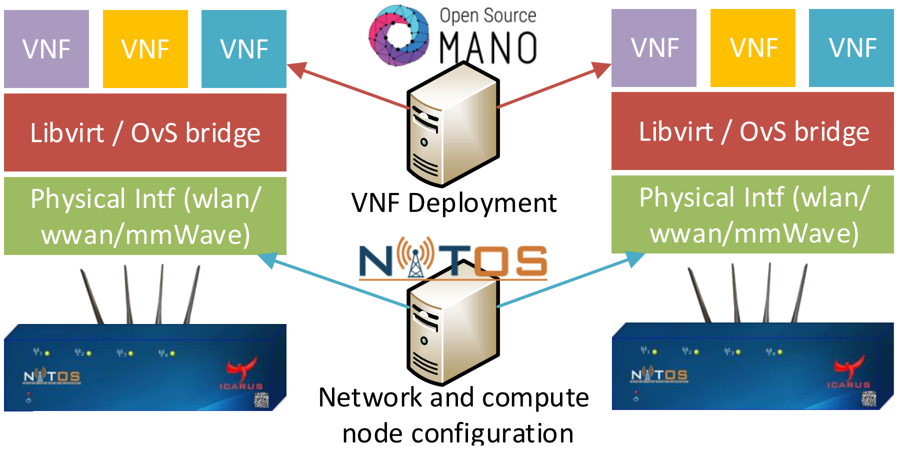


FIGURE 6.1: VNF instantiation on a NITOS node: each VNF is bridged to the underlying physical wireless network (WiFi/LTE/mmWave), configured through the NITOS testbed tools.

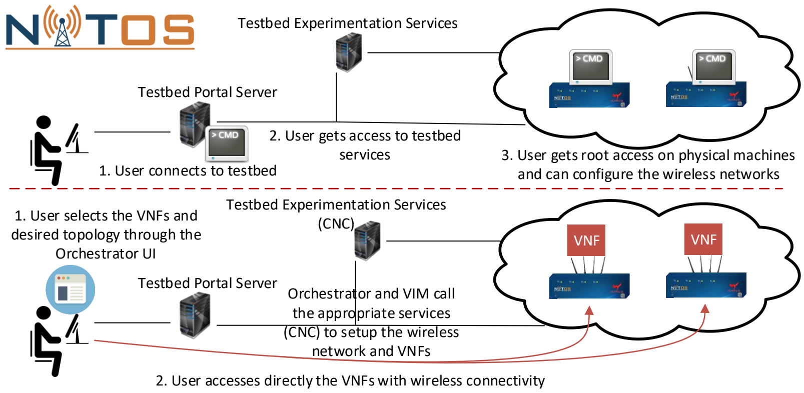


FIGURE 6.2: Existing vs. developed methodology for experimentation in the testbed; users do not have to learn the testbed's API for experiments, but just select the services they want to deploy and they type of connectivity they want. After the orchestration process, they get access directly to the VNFs.

6.4 System Architecture

The fact that traditionally NFV-MANO has been developed for orchestrating services and functions over datacenters hinders us from considering VNFs with wireless network services being deployment over them. Nevertheless, the structure of a testbed can be considered as a distributed set of nodes managed by an infrastructure manager such as Openstack [97] or OpenVIM [76].

The proposed contributions change radically the manner and complexity through which the testbed resources are provisioned and managed. Through the existing methodology for testbed access, users connect to the testbed, and using the offered services get superuser access on the physical machines. From this point on, they can configure the nodes based on their experimentation needs; they can install new software, setup wireless networking between them, etc. Through the developed methodology, the users just access the UI of the orchestrator software in the testbed.

They may use existing preconfigured VNFs, or upload their own, and specify at the VNF descriptor how these VNFs shall be connected for the wireless network as well. Beyond this point, the developed tools are in charge of configuring the wireless networks on the compute machines, and preparing them for hosting the VNFs. The experimenters eventually get direct access to the VNFs which also have wireless connectivity. Figure 6.2 presents this difference, and in this section, we further detail this process.

AAs all of the testbed nodes also use Ethernet connections, we use them as the management plane of the orchestrator. Since orchestrator software such as OSM usually refers to the configured VIMs as datacenters, we adopt the term for the set of the testbed nodes that we manage through our extensions. For the differentiation of the services deployed over the nodes, we organize the testbed in four different "datacenters"as follows:

- **SDR datacenter**: all the nodes with SDR capabilities are included in this datacenter. VNFs/PNFs related to the execution of services interfacing SDRs are using this datacenter. Example VNFs scheduled for this datacenter may include OpenAirInterface RAN VNFs, for setting up the eNodeB part of an LTE network.
- **LTE datacenter**: all the nodes that have LTE dongles are included in this datacenter. All the services using the LTE network to transmit data are using this datacenter.
- **WiFi datacenter**: all the nodes with WiFi connections are included in this datacenter. Similar to the previous one, all the VNFs/PNFs deployed can use a WiFi connection.
- **Ethernet/mmWave datacenter**: the rest of the nodes that use Ethernet connections. Through the Ethernet network, the nodes are able to use the mmWave equipment.

Depending on the type of services that will be deployed, we plan the deployment of the VNFs on top of the physical nodes as shown in Figure 6.1. The VNFs (Virtual Machines with ready to provision services) are making use of a bridged Ethernet connection with the physical interface that transmits the data over the air. Depending on the type of the physical interface used, we employ bridges based on either *bridge utils* or Open-vSwitch [87]. The VIMs of the testbed which have been extended are OpenVIM [76] and Openstack [97]. We select to extend both VIMs as the former provides a more lightweight approach to infrastructure management, and can be easily setup even on resource constrained servers. For deployments resembling a more "datacenter" oriented approach, we use Openstack. We make use of existing services for the configuration of the LTE equipment (*bscontrol/LTErf* [**lterf**]), or the automatic configuration of the *hostapd* instance for creating WiFi Access Points in the testbed.

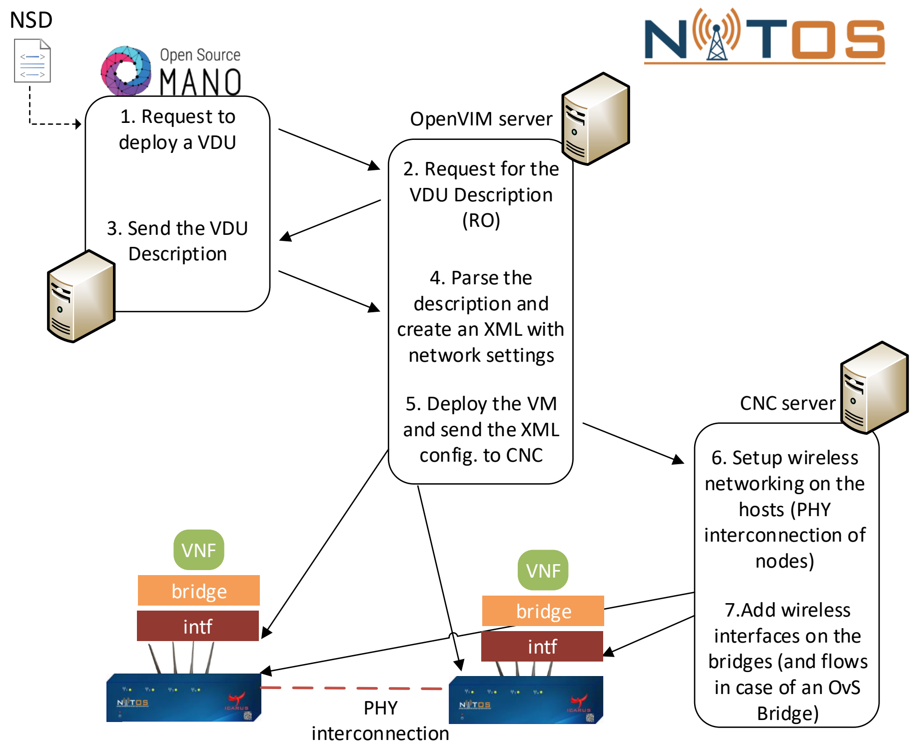
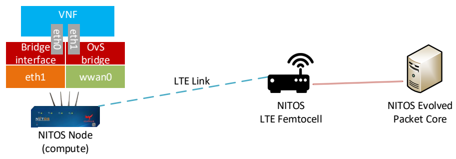
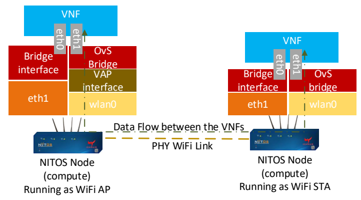


FIGURE 6.3: Intercommunication of different components for deploying VNFs over the wireless interfaces of the testbed

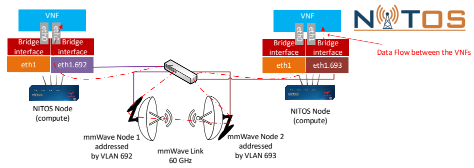
Figure 6.3 shows the actions taken for establishing the wireless connections in the testbed prior to the VNF instantiation. Initially, when OSM receives a Network Service Description (NSD) parses the different components, and for each VDU, sends a request to the VIM instance. For setting up the wireless connections in the testbed, we have included some extra information on the description field of the VDU. Listings 6.1, 6.2, 6.3, 6.4 show the respective configuration for setting up a VDU with LTE, WiFi or mmWave connectivity. When the VIM receives a request to deploy a VDU, asks the REST API of the OSM RO component about the description of the VDU. Upon the reception of this information, the data for the configuration of the network is parsed and kept at an XML file. This XML file is then sent via an HTTP POST to our service, the **Compute Network Control (CNC)** service, which is communicating with all the compute nodes. By introducing this service in the overall architecture, we provide the appropriate extensions for configuring our equipment in an automated and seamless manner, and preparing the compute nodes to host the VNFs. The CNC service starts to setup the wireless network on the compute nodes, and add the network interfaces to the respective bridges for networking the deployed VNFs. The information that we pass to the instance is through the description field of the VDU, as this information is not parsed by VIMs. This allows us to easily move VNFs between different VIMs as the information model for the communication between different components is remaining unchanged. The respective actions for provisioning the VNFs with wireless connectivity per each technology are described in the subsections below.



(A) VNF deployment with LTE connectivity over the COTS NITOS LTE network



(B) Pair of VNFs deployment with WiFi connectivity (one of them as an AP, the other as a Station)



(C) Pair of VNFs deployment with mmWave connectivity (point-to-point connection of physical nodes)

FIGURE 6.4: Deployment of VNFs over the NITOS compute nodes: physical interfaces are either bridged or routed to the VNFs for relaying all the traffic over the wireless networks

6.4.1 VNFs with LTE connectivity

For the LTE case, provisioning a VNF with such capabilities equals to providing a VNF with an extra networking interface that is bridged on the compute node with the physical interface connected to the COTS LTE network of NITOS. This process is depicted in Figure 6.4a. Prior to deploying the VNF, the CNC service takes care of turning the LTE dongle on the node on, and sending the appropriate AT-commands in order to connect it to the network.

The VNF that is deployed in the testbed, is using two different interfaces: one bridged to the management Ethernet network, used for accessing/controlling/bootstrapping the VNF, and a second one relaying the traffic over an LTE link. The bridge upon which the VNF interfaces are attached is OvS based; the CNC framework takes care of updating all the flows in the host machines involved in an NSD deployment, in order to enable their intercommunication. As the VNFs inside an NSD are instantiated in a sequential manner, CNC takes care of all the flow setup parameters for the existing and about to deploy VNFs. This means that for each subsequent VNF that is deployed, CNC revisits already deployed VNFs and updates the flows at the OvS bridge in order to enable the intercommunication among all the VNFs included in the NSD. Services running inside the VNFs, can take advantage of this link, to communicate with other VNFs connected to the LTE network, or run in a completely isolated manner from other LTE clients.

```
// Rest of VDU configuration omitted
...
    vdu:
```

```
  - cloud-init-file: cloud-config.txt
    count: '1'
    description: type:LTE,virtual-apn:APN1,dlambr: 50000000,
                  ulambr:50000000,qci:5,
                  ip:192.168.4.1,external-cp:lte-data
    id: LTEUEVM
    image: ...
...
// Rest of VDU configuration omitted
```

LISTING 6.1: Sample of VDU configuration for a LTE AP: the client that will be attached to APN1 will get at max 50 Mbps for DL and UL traffic

6.4.2 VNFs with WiFi connectivity

For the WiFi case, each node of the testbed is using a highly configurable WiFi card, can supports all of the modes of the IEEE 802.11 suite of protocols (Access Point, Station, Mesh, Ad-hoc, Monitor). For deploying the VNFs, only the modes of Access Point and Station are exposed through the NITOS OpenVIM instance. This means that experimenters may use the NITOS nodes in order to set them up as WiFi Access Points (APs), and use the respective nodes configured as clients and associated with them.

```
// Rest of VDU configuration omitted
...
  vdu:
  - cloud-init-file: cloud-config.txt
    count: '1'
    description: type:WiFiAP,essid:VWLAN1,ip:192.168.10.1,
              mode:g,channel:1,external-cp:wifi-data
    id: WiFiAPVM
    image: ...
...
// Rest of VDU configuration omitted
```

LISTING 6.2: Sample of VDU configuration for a WiFi AP: multiple can be instantiated over the same node with the *CNC* service taking care of the low level networking and bridging configuration.

Similarly to the LTE case, the VNF has a second interface which is routed through the WiFi network that is setup on the compute node. For the case that a VNF is deployed over an AP link, a virtual Access Point (VAP) is created through the hostapd tool [72] on the compute node. The VAP uses an ESSID (network name) that is configured from the experimenter side through the description field in the VDU description. Similarly, an experimenter may deploy one or more VNFs configured as WiFi stations (STA) that communicate over the specified ESSID. The deployed VNF has again two different interfaces, one used for managing the VNF and the traffic sent over the second one is bridged with Open-vSwitch.

```
// Rest of VDU configuration omitted
...
    vdu:
    - cloud-init-file: cloud-config.txt
      count: '1'
      description: type:WiFiClient,essid:VWLAN1,ip:192.168.10.2,
                 external-cp:wifi-data
      id: wifiClientVM
      image: ...
...
// Rest of VDU configuration omitted
```

LISTING 6.3: Sample of VDU configuration for a WiFi station associated with a (virtual) WiFi AP. The VNF deployed through this description will associate with the VNF deployed through Listing 6.2

The CNC service is in charge of setting up the appropriate rules on each of the nodes involved in a WiFi experiment, by configuring the flows on each OvS bridge (AP nodes and Station nodes). The flow rules regard the operations of changing the MAC addresses for the communication between the nodes for the ARP and data packets, so as the communication between all wireless stations and the AP is enabled by using the standard IEEE 802.11 frames. An overview of a pair of nodes using the WiFi link is shown in Figure 6.4b.

6.4.3 VNFs with mmWave connectivity

For the mmWave case, each NITOS node is able to use one the six mmWave nodes that are available in the testbed. The mmWave nodes support only point-to-point communication, and up to three pairs can be configured for simultaneous transmissions. Each of the six nodes is addressed through a VLAN interface; traffic sent over VLAN 692 is reaching the first node, VLAN 693 the second, etc. For the provisioning of VNFs using a mmWave link, a similar process is followed like for the previous technologies.

```
// Rest of VDU configuration omitted
...
    vdu:
    - cloud-init-file: cloud-config.txt
    count: '1'
```

```
    description:  type:mmWave, physical-node-id:1,
                  ip: 192.168.5.1, external-cp:mmWave-data
    id: mmwaveOneVM
    image: ...
...
// Rest of VDU configuration omitted
```

LISTING 6.4: Sample of VDU configuration for a mmWave node: all the underlying VLAN configuration is handled by the NITOS *CNC* tool

A VLAN interface is created on the compute node by CNC, attached to a bridge communicating with a second network interface of the VNF. Traffic sent over this interface gets subsequently transmitted over the VLAN, reaching the mmWave node and then over the air. At the receiving mmWave node, the traffic is encapsulated in the respective VLAN addressing the mmWave node. Traffic is transmitted over the VLAN and can be delivered at the VNF that has this VLAN configured at the compute node.

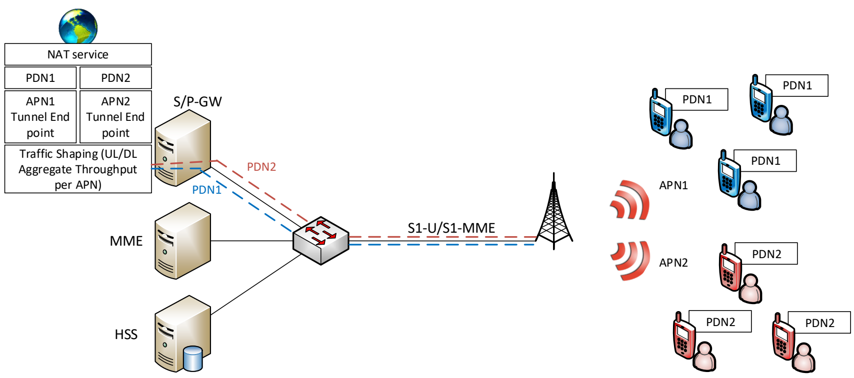


FIGURE 6.5: LTE slicing based on PDNs in the NITOS testbed: each PDN equals to a unique path from the core network (P-GW component) to the wireless RAN, through which only clients belonging to the same PDN can communicate. Each PDN can be set to exchange a value of a maximum UL/DL aggregate traffic for all the clients belonging to it.

6.5 Wireless Network Slicing

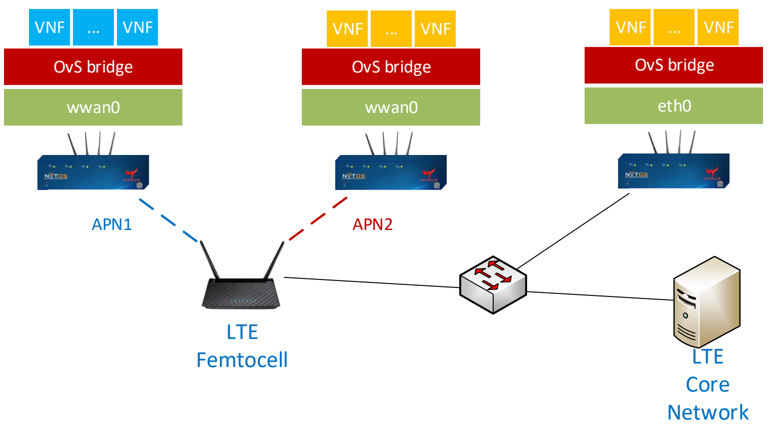
In this section we present our approaches in slicing the wireless networks that we use to bridge our VNFs to. The approaches to slicing are based on what can be applied over the COTS equipment that we use in the testbed. The following subsections show the practices applied to the 4G network equipment (LTE), WiFi and mmWave networks.

6.5.1 LTE Network Slicing

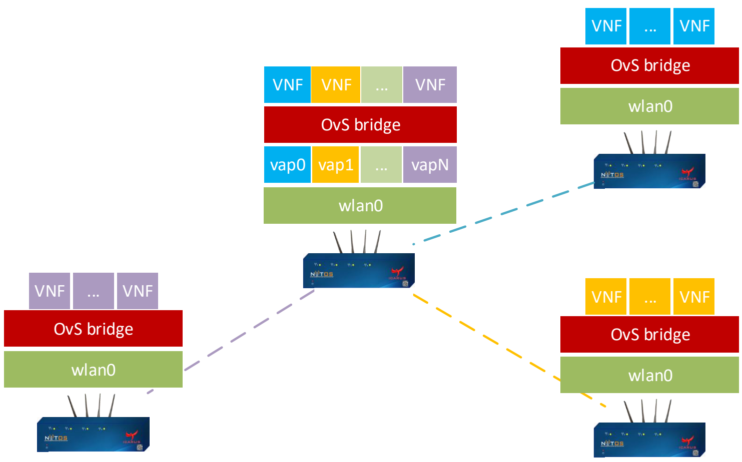
Although there has been significant work on the RAN based slicing for LTE (e.g. [59], [50]), yet, no commercial products seem to integrate such functionality. For the NITOS case, we employ the off-the-shelf LTE infrastructure that is offered. The RAN is provided by two ip.access small-cells, whereas the Core Network is a network-in-a-box solution by SiRRAN Communications [99]. Our high level target in slicing the network with a holistic approach (considering the RAN and Core Network as one entity) and to provide guarantees for different users on the usage of the network resources offered by the infrastructure. For this purpose, we exploit the notion of different Packet Data Networks (PDNs) to achieve this functionality, as follows: each PDN is realized as a separate tunnel for user plane traffic, running from the PDN-GW component of the Core Network (Evolved Packet Core - EPC) to each network UE. For the RAN case, each UE is using an Access Point Name (APN) to connect to the network. Each APN is the anchor router to each of the PDNs implemented. For the case of the NITOS testbed and the commercial Evolved Packet Core (EPC) that we are using [99], the anchor router is collocated within the same service as the Core Network. When a UE requests to connect to the LTE network, the RRC messaging exchange is containing the APN details that the UE wishes to connect to. These messages are exchanged between the UEs and the EPC. If the information is valid, and such a PDN is configured in the infrastructure, the UE is admitted to the network. The user plane data (data the UE is sending over the network) are then transferred from the base station to the EPC and vice-versa over dedicated GTP tunnels. The GTP tunnels are aggregated to a single interface on the core network side, per each PDN.

Each PDN is a separate broadcast domain for the network, and the UEs of the network use addresses belonging to this domain. Since all the clients of the LTE network are from the EPC's perspective interfaced by single tunnel interfaces per each PDN, we are able to isolate the flows of each PDN. Although UEs may be associated to the same base station, and their traffic is traversed through the same EPC, the operate in an isolated manner if they belong to different PDNs. Using this functionality, we can throttle the traffic that is exchanged over the network per each PDN from the EPC side; all the exchanged user traffic is traversing the EPC, even if a UE is trying to reach a nearby UE. This throttling is based on the maximum Uplink (UL) and Downlink (DL) traffic that the clients in each PDN exchange over the LTE network. An illustration of this slicing functionality is depicted in Figure 6.5. In the illustration, the blue (PDN1) and the red (PDN2) clients can only communicate with clients belonging to their own PDN. From the Core Network side, the traffic is throttled between the two different APNs, from the PDN-GW. Figure 6.6a illustrates the orchestration of VNFs by defining different APNs in the service descriptions. VNFs belonging to interfaces associated with different APNs will not be able to communicate with each other. Nevertheless, assigning a new VNF that belongs to the same

APN (if connected through the LTE technology) or PDN (if using Ethernet as in the illustration), will enable the intercommunication with the respective VNFs (e.g. the yellow VNFs at the illustration).



(A) Orchestration of VNFs belonging to specific slices of the LTE infrastructure. VNFs are orchestrated through OSM, whereas the LTE physical interconnection and bridging with the VNFs is offered through the *CNC* service. VNFs are able to communicate only with the clients belonging to their own PDN.



(B) VNF instantiation over WiFi links in the testbed: VAP creation and physical interconnection is provided by the *bscontrol service* whereas the *iptables* rules on the host controlling the VAP instances ensure that VNFs on the clients can communicate only with VNFs belonging to the specific VAP they are associated to.

FIGURE 6.6: VNF Orchestration with multiple slices with WiFi and LTE connections.

6.5.2 WiFi Network Slicing

For the WiFi case, we employ the *hostapd* service to setup wireless Access Points (APs). Through its configuration, *hostapd* can setup over a single physical card multiple AP instances, known as virtual Access Points (VAPs). Each VAP has separate settings for the transmitted ESSID parameter, which the end user devices employ to associate to the network. Physical network parameters, such as transmission power and channel configuration are the same across different VAPs. However, by using the separate data queues for the different types of data (Voice, Video, Best-Effort and Background) that the WiFi drivers have, prioritization between different VAP instances can take place, as it has been shown in other works, e.g. [61], [110]. However, given the Listen-Before-Talk (LBT) protocol of WiFi, all the end-users compete in order to get access to the medium. Hence, contrary to the LTE case, no guarantees can be given through these methods on the per-VAP exchanged traffic. Nevertheless, these methods regulate the probabilities of each VAP to access the channel in a percentage of the total available capacity, constrained by the external interference that is present in the served region. Figure 6.6b depicts an example of deploying sets of VNFs over nodes configured as WiFi APs or STAs. Only the nodes within the same wireless network domain can communicate with each other (e.g. the blue VNFs on the STA node can communicate only with the blue VNF on the AP node).

6.5.3 mmWave Node Slicing

For the case of the mmWave nodes employed in NITOS we use the following approach. Each of the nodes is terminated to an OpenFlow switch, communicating over a specific VLAN. Different VLANs are allocated for each of the nodes. The termination switch of the mmWave nodes is connected to an other set of OpenFlow switches that interconnect the rest of the nodes of the testbed. All these switches can be programmed with individual controllers that each experimenter can setup, by slicing the switches using the FlowVisor tool [98]. For each user controller, only the ports that communicate with the reserved testbed nodes in this testbed account are included in the configurable flow space of the end user. On the mmWave nodes (the physical nodes which offer the mmWave connections to the testbed nodes), OpenvSwitch (OvS) is used to bridge the Ethernet VLAN interface with the actual radio interface of the node. This gives us the advantage that all the nodes of the testbed can use the mmWave network connections, by setting up the correct VLAN interface on the node and enabling the respective VLANs on the OpenFlow switches. These VLANs are configured through the CNC service, that manages all the low level configurations needed on the nodes such as creating the appropriate VLAN interface addressing each mmWave node, setting MTU settings on this interface, etc. Although currently no guarantees are provided for each slice airtime, the OvS instance that is running natively on each mmWave node allows for dynamic scheduling of the flows by making use of the Hierarchical Token Buffer (HTB) switches [44] of the Linux kernel. Per client identification of flows is planned to be supported through the use of IEEE 802.1Q-in-Q VLANs; the outer VLAN tag is used to identify the mmWave node that will be used for transmitting, whereas the inside VLAN tag is used to identify each tenant (VNF) that is making use of it. Load balancing or scheduling each of the flows relies on establishing match actions for the inner VLAN tag on each mmWave node.

6.6 Testbed Integration

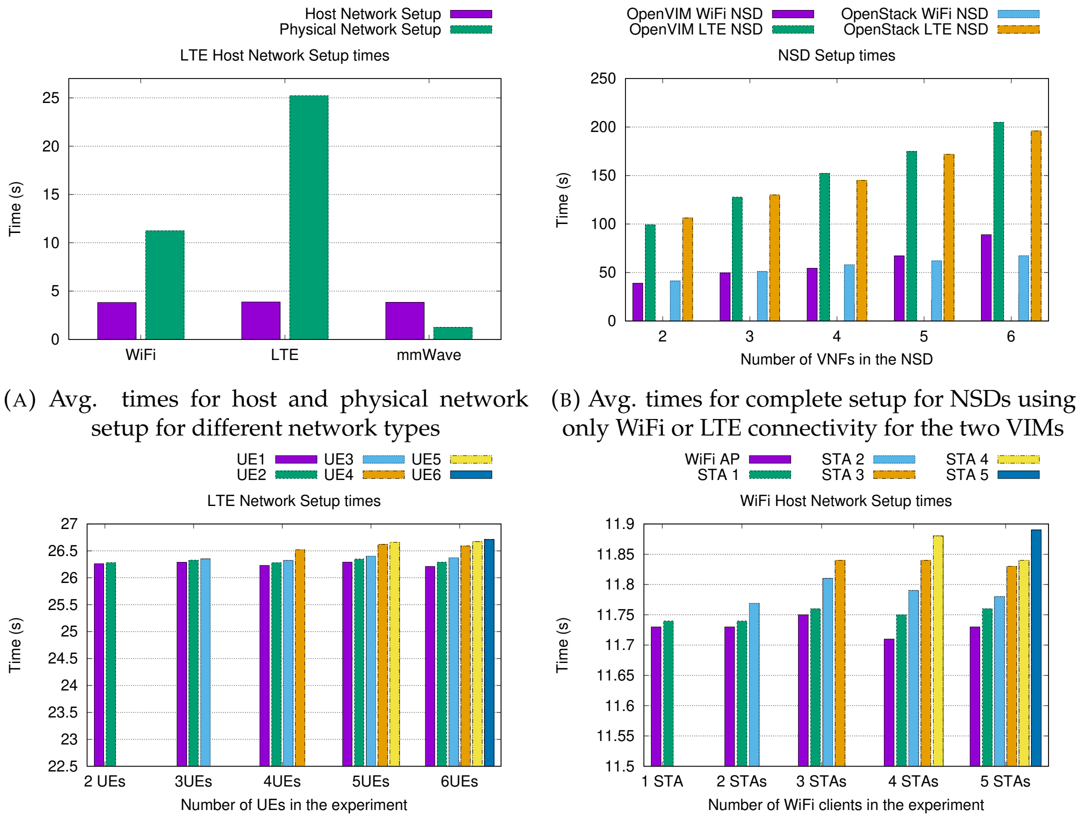
In this section, we detail how the aforementioned schemes have been integrated in a testbed environment. Prior to our contributions, experimenters using the testbed have to go through the process of learning the testbed API, and make extended use of command line tools in order to setup a simple experiment. Through our contributions, we are able to provide the users with the orchestrator UI, and allow them to deploy their experimental topology as VNFs over the testbed. By adopting the extensions presented in the previous sections, they can specify the wireless links between the VNFs, and effortlessly deploy even large-scale experiments in a single-click manner. The target testbed where we apply our extensions is NITOS testbed. NITOS is a wireless oriented experimentation facility, established under the Future Internet Research and Experimentation (FIRE) initiative [37]. Throughout the years, FIRE established a set of tools and methodologies for experimenting with testbed resources. These include schedulers for moderating the access to physical or virtual resources, APIs for orchestrating experiments, measurement methodologies and tools for creating larger facilities through the federation of isolated testbed islands. Since one of our goals is to provide the testbed resources with the NFV-MANO approach, these extensions regard the experimental plane of the testbed (i.e. the process of actually orchestrating the experiment). The integration with the existing tools described hereunder regards the control plane part of the testbed (i.e. preparing the resources for experimentation).

The fact that FIRE testbeds target repeatable experimentation over heterogeneous technologies (even in the wireless domain) has led to the adoption of a scheduler interface [101], where experimenters can select the number of nodes they want to use within a timeslot. Timeslots for experimenters in NITOS are at maximum four hours per reservation, with half hour slots. This calendar based reservation is very common within the FIRE testbeds federation, such as Fed4FIRE [105] and OneLab [31], especially for wireless resources. The existing mechanisms are based on the Slice-based Federation Architecture (SFA) protocol [86], which represents the resources of the testbeds with XML documents, referred to as Resource Specifications (RSpecs).

Since OSM assumes that the underlying compute resources are available for instantiating services at the time of the orchestrator, we developed the following methodology and tools for reserving the testbed resources prior to the orchestration. Before the orchestration takes place, a daemonized service running at the OpenVIM side is querying the REST API of the OSM Northbounf Interface (NBI) for any new onboarded VNFs. This service is running in an entirely isolated manner from the rest infrastructure, in a container, and processes the JSON responses from OSM. In case that there are any new VNFs onboarded to OSM, and hence an experiment is about to be orchestrated, the service starts the reservation process in the NITOS testbed by communicating with the NITOS broker [101]. Based on the parsed data, the exact physical resources (compute) that will be used for a specific experiment can be concluded on the NITOS side, and therefore reserved as follows. In case the VNF descriptions of the onboarded NSDs contain the extra information that is depicted in Listings 6.1, 6.2, 6.3 or 6.4, the respective nodes are filtered from the broker response and requested for reservation. In case that an SDR node is requested, this is provisioned by indicating the USB controller as a pass-through device. The communication with the broker is taking place over the Slice Federation Architecture (SFA) [86] protocol (Versions 2 and 3 are supported), that is a standardized interface for FIRE testbeds.

Upon the conclusion on the exact technologies, and therefore compute nodes, that shall be used for an experiment, the daemon process makes the respective calls to the NITOS broker for retrieving the available nodes within a specific timeslot for four hours from the point that the VNFs are onboarded. If the nodes are available, the tool reserves them and places them under a single slice name per each reservation. This shall not be confused with the notion of slices in virtualized infrastructure, but refers to a set of physical resources in the testbed. The selection of the compute nodes relies on the network interfaces that the nodes have (e.g. LTE, WiFi, SDR devices) and is accomplished through a catalogue system that the service maintains and filters the sets of nodes that can be used for each specific experiment. A job is also registered to take place before the experiment orchestration begins, by preparing the physical resources and loading the appropriate image on them by querying the REST API of the NITOS Broker. The nodes are also added to the NITOS VIM in order to allow the instantiation of VNFs on top of them. This image refers to a binary copy of the hard disk of a node, including the operating system, and all the kernel modules, services and applications that will run on it. The image is currently an Ubuntu Server 14.04 installation, with the *virsh* and *Qemu 3.0* services installed, and some scripts for connecting the LTE dongles to the network or configuring the node as a WiFi AP (e.g. *hostapd* configuration with multi-VAP mode, etc.).

When the experiment ends, the VNFs are deleted from the OSM instance of NITOS; this allows us to re-orchestrate the same VNFs in a subsequent experiment instance. The developed functionality is generic enough to be applied as it is for any other NITOS-like testbed that may integrate such an approach for reserving the testbed's resources. Moreover, as the description field in the VDUs that we use for passing the information to the VIM instance is not parsed by OSM, the same VNFs can be deployed at any other OSM instance that is not using our extensions; in such a case, the experiment will be orchestrated with Ethernet connectivity by default.



(A) Avg. times for host and physical network setup for different network types

(B) Avg. times for complete setup for NSDs using only WiFi or LTE connectivity for the two VIMs

(C) Avg. times for host network setup for VNFs using the LTE network

(D) Avg. times for host network setup for VNFs using the WiFi network

FIGURE 6.7: Benchmarking results for the orchestration process

6.7 System Evaluation and Bechmarking

For the evaluation part of the proposed integration with the testbed, we evaluate the framework by means of two different experiments: 1) a benchmarking, measuring time for deployment of different types of VNFs with wireless connectivity in the testbed, and 2) a use case experiment, dealing with the deployment of wireless base stations, where the base stations are instantiated as VNFs using our framework. In the following subsections, we detail our experimental findings.

6.7.1 Framework Benchmarking

For the benchmarking part of our evaluation, we select different scenarios of deployment of NSDs in the testbed, for provisioning VNFs with wireless connectivity. We measure the following parameters:

1. the *host network setup time*, which is the time that the CNC tool needs from the time that it is invoked from the testbed VIM to setup the network bridges on the host machine where the VNFs will be instantiated,
2. the *physical network setup time*, which is the time that is required for the actual physical network interfaces to connect to the network. For example, the physical

setup time for the LTE technology is the time needed for the host machine to connect its LTE dongle to the LTE network of the testbed.

3. the *NSD deployment time* for orchestrating VNFs with wireless network interfaces. This is the time since the orchestration command is issued in the OSM dashboard until the VNFs are deployed and available in the testbed. For this part of evaluation, we examine the performance of both VIMs that we integrated our changes into, namely OpenVIM and Openstack, as an effort to directly compare the two solutions over exactly the same conditions.

Figure 6.7 presents our experimental results for the aforementioned indicators. Figure 6.7a presents the average times for the host network and physical network setup for the different types of wireless connectivity. These times are concluded when deploying a NSD with two VNFs, communicating over the same technology. As we see, all the host network setup times are almost the same; this is regular, as the time that we measure regards the period for configuring the network bridges and attaching the physical interfaces. This process is straightforward and identical for all the wireless networks. Contrary to this, the physical network setup times regard the time needed for setting up the wireless connectivity on the host machines; this for the case of WiFi connectivity reflects on bringing up the wireless driver on the host node, setting up the Access Point configuration, address setup between the nodes, and setting up the network flows on all the physical nodes involved in the experiment. As we see, for the case of mmWave connectivity, the times are very low compared to other technologies. This is due to the fact that for setting up a mmWave connection in the testbed, we only need to configure a VLAN interface on the host machine, and configure parameters such as the beam angle on the mmWave node that will be used.

Figure 6.7b shows the comparison between the overall times for setting up an NSD with up to 6 VNFs using LTE or WiFi connections, when using the two extended VIMs (OpenVIM and Openstack). As we see, setting up LTE connectivity for the VNFs takes significantly more time than WiFi, due to the complexity of setting up each connection; CNC tool needs to connect to each of the host nodes and issue the appropriate AT-Commands. However, we note a difference between the two VIMs for the deployment times subject to the number of VNFs included in the Network Service Descriptor; for up to 3 VNFs, we observe that OpenVIM is able to deploy the VNFs in a faster manner. When more VNFs are employed (e.g. 4-6), Openstack is faster in the total deployment time by a few seconds. It is worth to mention that the times are measured until each and every VNF in the descriptor starts running, and not when it reports back to the orchestrator; here we observed another difference of the tools, as Openstack notifies the orchestrator as soon as the VNFs have been assigned IP addresses, while OpenVIM spends approx. 40 seconds before reporting back. This is an issue that we plan to investigate in our future works.

Figures 6.7c and 6.7d show the physical network setup times for varying number of VNFs in the submitted NSD, for LTE and WiFi connectivity respectively. As we see, the time is growing linearly as the number of VNFs in the NSD rises (UEs or WiFi Stations). This is due to the additional process of the flow configuration on each node, each time that a new host is added to the wireless network. In such a case, the CNC tool revisits all the configured nodes and updates the flows on each one of them Finally, Figure 6.7b shows the overall times for setting up an NSD with up to 6 VNFs for LTE or WiFi connectivity.

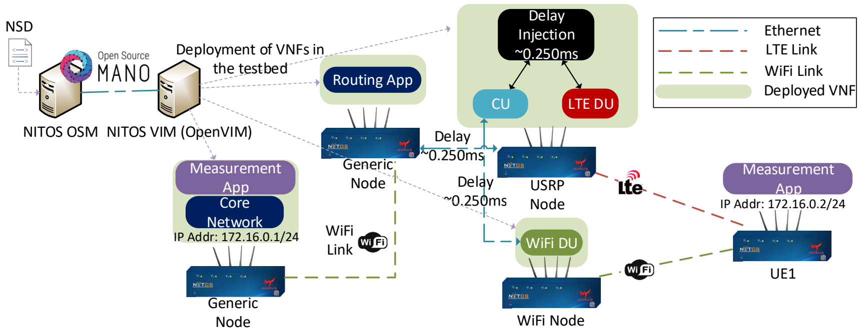


FIGURE 6.8: Deployment architecture for disaggregated heterogeneous Cloud-RAN base stations as VNFs in the NITOS testbed: wireless interfaces (SDR and WiFi card) are configured as pass-through devices for the access network, whereas for the backhaul network we use our extensions to configure a WiFi link. Depicted delays have been measured over the links interconnecting the nodes of the testbed. For the delay injection case, we use the *netem* application. [41]

6.7.2 Wireless Base Station deployment

For the second part of the evaluation, we employ a software base station and orchestrate the operation of a Cloud-RAN, integrating heterogeneous technologies on the RAN. The work has been described in [65] in detail, and describes the functionality added to the OpenAirInterface (OAI) platform [83] in order to define a data plane split for the LTE stack, and integrate heterogeneous technologies for the cell. This work considers the split of the base station stack at the high Layer 2 of the OSI stack, between the Packet Data Convergence Protocol (PDCP) layer and the Radio Link Control (RLC) layer. The resulting operation of the base station stack splitting has two outputs, based on the specifications for the 5G stack [**f1-arch**]: the Central Unit (CU), incorporating the higher layer functionality and the interface to the Core Network, and the Distributed Units (DUs) incorporating the lower layers including the transmission/reception of data over the air. As the OAI platform is purely based on software, it exemplifies how softwarization can extend even to the RAN, and how orchestration can include such services as well. In this use case, we orchestrate the operation of such a network through our extensions. For the orchestration of this experiment, we focus on OpenVIM, as it provides a more simple solution for passing through the USB SDR devices. Openstack on the other hand, requires user involvement after passing-through the USB controller hosting the SDR from inside the VNF, as supplementary to a PCI pass-through of the USB controller, requires USB pass-through for the device as well. As OpenVIM supports both features in a seamless manner, we select this to host our experiments. We deploy four different VNFs for the wireless network service as follows:

- A Core Network VNF, running the OpenAirInteface Core Network software.
- A RAN VNF, running the CU and DU functionality for the LTE part of the network, including the interface for heterogeneous DUs.
- A WiFi DU VNF, in charge of transmitting/receiving data to/from the WiFi UEs of the network from/to the CU side.
- A simple VNF that routes the data between the CU and the Core Network.

For the DU sides of the network, we use PCI pass-through for the LTE DU VNF, in order to attach the entire USB3 controller that hosts the SDR device needed for transmitting data over the network. We employ a similar approach for the WiFi DU that runs our software, but handles the modes of the wireless card. Hence, we pass-through the PCI device that holds the WiFi card of the network. The WiFi DU part of the network encapsulates data received from the UEs in its own format (details on the framework are shown in [65]) and transmits them to the CU. The reverse process is done for downlink traffic. We use the extensions in the orchestrator in order to support a WiFi based path in the backhaul network (between the routing VNF and the Core Network). The topology for the deployed VNFs is shown in Figure 6.8. We depict the VNFs and reported delay for their intercommunication in the testbed. All the nodes are synchronized using the NTP protocol [77], providing millisecond based accuracy among the clocks of the testbed's nodes.

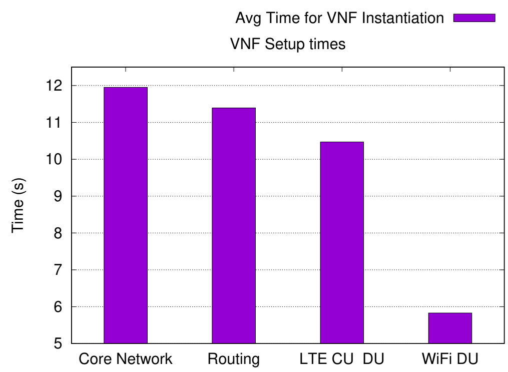
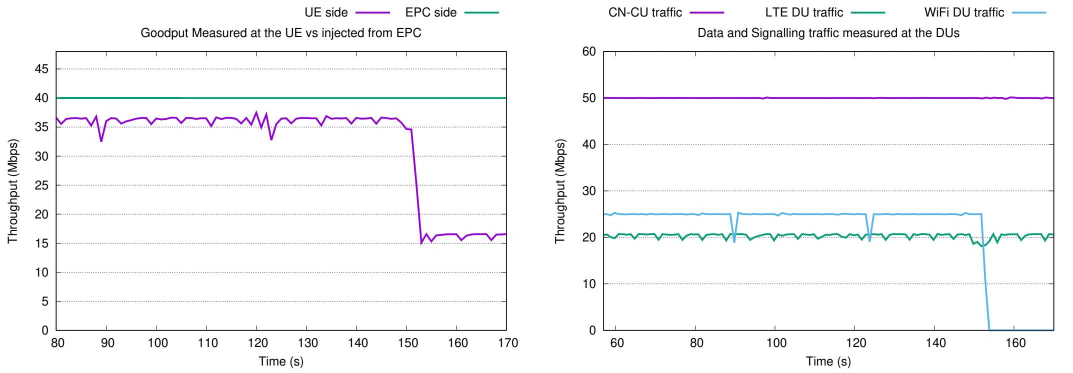


FIGURE 6.9: Average Setup time for the VNFs in the use case experiment for heterogeneous Cloud-RAN base station instantiation.

Figure 6.9 illustrates the respective times for the instantiation of each VNF involved in the service descriptor. The entire time for deploying the NSD is approx. ~ 96 seconds, averaged from 20 experiment runs. As we see, the two VNFs that are using our extensions for WiFi connectivity (Core Network and Routing VNFs) take



(A) UE goodput received from the two networks, when injecting 40Mbps at the Core Network

(B) Signaling and traffic transferred over the two DUs of the system.

FIGURE 6.10: Experimental results for the base station deployment scenario.

approximately the same time as for the VNFs that we showcased in the previous subsection. During these times, the setup of the wireless network takes place as well, by using the Cloud-Init framework in order to launch the network operation software. The WiFi DU and LTE network VNFs take less time, as they require only a pass-through connection for each device they are handling. Following this initial benchmarking, we test the functionality of the VNFs by attaching a multihomed UE to the network (WiFi and LTE). We inject 40Mbps of traffic from the Core Network side using the *iperf* application, with the CU being configured to split the traffic over both DUs (LTE and WiFi). It is worth here to mention, that we use a SISO configuration with 5MHz of bandwidth for the LTE part of the network, which in all of our testbed experiments may wield $\sim$ 16.5Mbps of goodput at the UE. Figure 6.10a shows how the traffic that is received by the UE versus the one injected at the Core Network, and Figure 6.10b the amount of traffic transferred over each DU of the network. At approx. 150 seconds of the experiment, we bring down the wireless interface on the WiFi DU, and see that the UE is getting the traffic transferred over the LTE DU only, equal to approx. 16.5Mbps.

6.8 Discussion and Chapter Conclusion

The provided functionality is able to accelerate the deployment of new services that might be running over different networks, towards their experimentally driven validation. The developed approach is a fully-fledged solution for application prototyping over testbeds, allowing with minimal overhead the experimenters to deploy them over a distributed datacenter with wireless technologies. Through the description of the applications in VNFDs, and the adoption of the extended syntax for the VNFDs allowing the specification of the network parameters, the presented approach is a solution fitting several needs. From the testbed administrator side, the solution is packed as a VIM exclusively for nodes that are interconnected and communicate with each other, and participate in forming a distributed datacenter solution. Thus, it can be easily installed in any other NITOS-like FIRE testbed. Moreover, for the testbed administrator case, this approach is able to provide enhanced portability of experiments over the testbeds, regardless of their type, as long as they can process YAML based service descriptions that OSM consumes. Providing such extensions to the testbed mildens the learning curve of using the infrastructure; the end-user is only presented with a repository/marketplace of VNFs, which can be instantiated from a higher layer by organizing and designing the interconnection of the components through OSM. Also VNFs developed/running at other sites can be easily instantiated in any testbed featuring such extensions.

In the future, we foresee extending the service orchestration in the testbed by making use of tools like JuJu, Ansible or Cloud-Init, for the seamless and effortless bootstrapping and on-boarding of services on the VNFs. Currently, we have experimented with the JuJu functionality in OSM Rel FOUR and FIVE, and allows us to get similar behaviour over the testbed as with the legacy experiment description and orchestration tools for physical machines (OMF). Through the development of proxy charms for JuJu, almost any application used for off-the-shelf experimentation can be controlled through the orchestration software. Processes that are based on software and usually take up a lot of time for the setup of the experimentation environment in the testbed, e.g. setting up a software based base station over a USRP device, can be fully automated, by the selection of the interfaces that will backhaul/fronthaul the base station from the top level, as shown in the second use case experiment.

Chapter 7

Conclusions and Future Work

In this thesis, we focused on the disaggregated architecture for base stations, for supporting future 5G cellular networks. Through the formation of Cloud-RANs, flexibility is added for the management and commissioning of new base stations, even on the fly, based on the current demand in an area. All of our contributions were experimentally driven, implemented in open-source software that runs over commodity hardware, in order to enable performance evaluation under realistic environment settings, and enable the direct comparison of our frameworks with existing solutions and standards.

As traffic and signaling for splits taking place after the baseband processing can be transported over a packetized fronthaul, we focus on higher layer 2 splits. Initially, we presented an experimental evaluation of different layer 2 splits and under different settings for the wireless RAN. The solution was developed over the OpenAirInterface platform and the overhead for different modulation and coding schemes and bandwidth settings was measured. This experimental evaluation was one of the first in literature which experimentally evaluates splits in the higher layer 2.

Subsequently, we explored how heterogeneous technologies can be aggregated in the cell. We used as a starting point what is considered in the previous technologies (e.g. through the LTE WLAN Aggregation feature) and integrate non-3GPP technologies to the network cell. We introduced them in the disaggregated Cloud-RAN when considering the split between the PDCP and RLC layers, as also considered in the NR specifications. The non-3GPP technologies are introduced as Distributed Units in the disaggregated architecture, thus enabling the dynamic selection of the technology that serves each multihomed UE in a per-packet basis. Our real-world experiments provided insights on the maximum distance that a DU (either 3GPP or non-3GPP) can be located from the (edge) datacenter instantiating the CU without degrading the performance that the end-user gets.

Following this, we focused on the formation of heterogeneous ultra-dense disaggregated networks and deal with the spectrum management for the under consideration DUs. As network densification is expected to play a key role in augmenting the wireless network capacity in an area, appropriate placement of the operating units is needed. In this context, and leveraging the disaggregated architecture, we designed, implemented and evaluated an algorithm for the efficient coordination of DUs in an area, depending on the presence of heterogeneous technologies. The algorithm relied on low-level statistics collected from all the DUs controlled through the same CU, which hosts the coordination process. For all the under-study cases, we observed that our scheme is more efficient than the existing off-the-shelf solutions, allowing higher capacity in the provided networks.

Building on the disaggregated heterogeneous base station architecture, we further studied the possibility of minimizing the service access latency for the end-users. Existing suggested deployments for Multi-access Edge Computing, which is considered to be the enabler for low latency communications in 5G, suggest the placement of services beyond the (cloud) base station, either on the backhaul network or beyond the core network. Using our prototype implementation, we designed, developed and experimentally evaluated our solution for placing services on the fronthaul network, closer or collocated with the network's DUs. Our evaluation demonstrated low latency for LTE access (below 10 msecs), creating fertile ground for 5G applications to be executed over legacy equipment.

Finally, we presented our contributions in orchestrating our software heterogeneous cloud-RAN base stations through the NFV MANO architecture. By exploiting the Open Source MANO tool, we extended how software services are deployed in a wireless testbed. Our contributions to the framework allowed the provisioning of VNFs making use of either WiFi, LTE or mmWave connectivity. Using as VNFs the disaggregated heterogeneous Cloud-RAN software, we are able to enable the single-click deployment of the mobile network, and provision a wireless backhaul/fronthaul in an effortless manner.

In the future, we foresee extending our schemes towards integrating NR functionality in the physical layer, as well as apply machine learning approaches for the cases of MEC and spectrum management; in the former, we plan to investigate possible dynamic placement of services on the fronthaul, based on the demand and some adoptive pricing policy that each network user might employ. For the latter, we plan to model the dynamic characteristics of the wireless network based on historical measurements, and through a machine learning approach to dynamically adapt the operating channel of the network's DUs.

Bibliography

[1] 3GPP. *"3GPP TR 38.806 V15.0.0 (2017-12), 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Study of separation of NR Control Plane (CP) and User Plane (UP) for split option 2; (Release 15)"*. 2017.

[2] 3GPP. *"3GPP TS 23.402 V14.7.0 (2018-03), 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Architecture enhancements for non-3GPP accesses (Release 14)"*. 2018.

[3] 3GPP. *"3GPP TS 36.360 V14.0.0 (2017-03), 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); LTE-WLAN Aggregation Adaptation Protocol (LWAAP) specification (Release 14)"*. 2017.

[4] 3GPP. *"3GPP TS 38.401 V0.2.0 (2017-07), 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; NG-RAN; Architecture description (Release 15)"*. 2017.

[5] 3GPP. *"3GPP TS 38.470 V0.3.0 (2017-09), 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; NG-RAN; F1 general aspects and principles (Release 15)"*. 2017.

[6] 3GPP. *"3GPP TS 38.473 V15.1.1 (2018-04), 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; NG-RAN; F1 application protocol (F1AP) (Release 15)"*. 2018.

[7] NGMN Alliance. "Further study on critical C-RAN technologies". In: *Next Generation Mobile Networks* (2015).

[8] I. Alyafawi et al. "Critical issues of Centralized and Cloudified LTE-FDD Radio Access Networks". In: *IEEE International Conference on Communications (ICC)*. IEEE. 2015.

[9] J. Bartelt et al. "Fronthaul and Backhaul Requirements of Flexibly Centralized Radio Access Networks". In: *IEEE Wireless Communications* 22.5 (2015).

[10] J. Bartelt et al. "Heterogeneous Backhaul for Cloud-based Mobile Networks". In: *78th Vehicular Technology Conference (VTC Fall)*. IEEE. 2013.

[11] J. Bastos et al. "Energy efficiency optimization for downlink OFDMA system in heterogeneous network with QoS constraints". In: *International Journal of Communication Systems* (2015).

[12] C. J. Bernardos et al. "5GEx: realising a Europe-wide multi-domain framework for software-defined infrastructures". In: *Transactions on Emerging Telecommunications Technologies* 27.9 (2016), pp. 1271–1280.

[13] D. Bernstein. "Containers and Cloud: From LXC to docker to Kubernetes". In: *IEEE Cloud Computing* 1.3 (2014), pp. 81–84.

[14] G. Bhanage, I. Seskar, and D. Raychaudhuri. "A service oriented experimentation framework for virtualized WiMAX systems". In: *International Conference on Testbeds and Research Infrastructures*. Springer. 2011, pp. 152–161.

[15] G. Bhanage et al. "Virtual Basestation: Architecture for an Open Shared WiMAX framework". In: *Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures*. ACM. 2010, pp. 1–8.

[16] S. Bhaumik et al. "CloudIQ: a framework for processing base stations in a data center". In: *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*. ACM. 2012.

[17] P. Biondi. *"Scapy: explore the net with new eyes"*. Tech. rep. Technical report, EADS Corporate Research Center, http://www.secdev.org, 2005.

[18] A. M. Cavalcante et al. "Performance Evaluation of LTE and Wi-Fi Coexistence in Unlicensed Bands". In: *2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*. IEEE. 2013, pp. 1–6.

[19] C. Y. Chang et al. "Impact of packetization and functional split on C-RAN fronthaul performance". In: *2016 IEEE International Conference on Communications (ICC)*. DOI: 10.1109/ICC.2016.7511579.

[20] A. Checko et al. "Cloud RAN for mobile networks - a technology overview". In: *IEEE Communications surveys & tutorials* 17.1 (2015).

[21] Visual Networking Index Cisco. *Global Mobile Data Traffic Forecast Update, 2015-2020 White Paper, 2016*. 2016.

[22] Y. Dai et al. "Interference control based on beamforming coordination for heterogeneous network with RRH deployment". In: *IEEE Systems Journal* 9.1 (2015).

[23] A. G. Dalla-Costa et al. "Maestro: An NFV Orchestrator for Wireless Environments Aware of VNF Internal Compositions". In: *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. Mar. 2017, pp. 484–491. DOI: 10.1109/AINA.2017.126.

[24] NTT DOCOMO. *"R3-161687, Draft TR 38.801 (v030) Study on New Radio Access Technology: Radio Access Architecture and Interfaces", NTT DOCOMO, INC (Rapporteur), 3GPP TSG RAN3, August 2016.*

[25] U. Dötsch et al. "Quantitative analysis of split base station processing and determination of advantageous architectures for LTE". In: *Bell Labs Technical Journal* 18.1 (2013).

[26] S. Dräxler et al. "SONATA: Service programming and orchestration for virtualized software networks". In: *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE. 2017, pp. 973–978.

[27] M. Emara, M. C Filippou, and D. Sabella. "MEC-aware cell association for 5G heterogeneous networks". In: *Wireless Communications and Networking Conference Workshops (WCNCW), 2018 IEEE*. IEEE. 2018, pp. 350–355.

[28] M. Ersue. "ETSI NFV management and orchestration-An overview". In: *Proc. of 88th IETF meeting*. 2013.

[29] ETSI. *"ETSI GS MEC 011 V1.1.1 (2017-07): Mobile Edge Computing(MEC); Mobile Edge Platform Application Enablement"*. 2017.

[30] ETSI. *Open Source MANO*. [Online], https://osm.etsi.org. 2016.

[31] S. Fdida, T. Friedman, and T. Parmentelat. “OneLab: An open Federated Facility for Experimentally Driven Future Internet Research”. In: *New Network Architectures*. Springer, 2010, pp. 141–152.

[32] Small Cell Forum. *“Document 159.07.02, Small cell virtualization functional splits and use cases, Small Cell Forum”, January 2016.*

[33] A. Francescon et al. “X-MANO: An open-source platform for cross-domain management and orchestration”. In: *2017 IEEE Conference on Network Softwarization (NetSoft)*. July 2017, pp. 1–6. DOI: 10.1109/NETSOFT.2017.8004236.

[34] A. Francescon et al. “X-MANO: Cross-domain management and orchestration of network services”. In: *2017 IEEE Conference on Network Softwarization (NetSoft)*. July 2017, pp. 1–5. DOI: 10.1109/NETSOFT.2017.8004223.

[35] J. F. Frye Jr. *Embedded OS PXE server*. US Patent 7,085,921. Aug. 2006.

[36] F. Fund et al. “Performance of DASH and WebRTC Video Services for Mobile Users”. In: *2013 20th International Packet Video Workshop*. Dec. 2013, pp. 1–8. DOI: 10.1109/PV.2013.6691455.

[37] A. Gavras et al. “Future Internet Research and Experimentation: the FIRE initiative”. In: *ACM SIGCOMM Computer Communication Review* 37.3 (2007), pp. 89–92.

[38] F. Giust et al. *ETSI White Paper No. 24: MEC Deployments in 4G and Evolution Towards 5G*. 2018.

[39] M.U. Hadi. “Extending the benefits of LTE to unlicensed spectrum”. In: *2015 International Conference on Information and Communication Technologies (ICICT)*. IEEE. 2015, pp. 1–3.

[40] Red Hat. *libvirt: The virtualization API*. 2012.

[41] Stephen Hemminger et al. “Network emulation with NetEm”. In: *Linux conf au*. 2005, pp. 18–23.

[42] G. R. Hiertz et al. “IEEE 802.11s: The WLAN Mesh Standard”. In: *IEEE Wireless Communications* 17.1 (Feb. 2010), pp. 104–111. DOI: 10.1109/MWC.2010.5416357.

[43] A. Huang et al. "Low latency MEC framework for SDN-based LTE/LTE-A networks". In: *Communications (ICC), 2017 IEEE International Conference on*. IEEE. 2017, pp. 1–6.

[44] B. Hubert et al. "Linux advanced routing & traffic control". In: *Ottawa Linux Symposium*. Vol. 213. 2002.

[45] *Hurtle project*. [Online], `http://hurtle.it/`.

[46] M. Jiang, M. Condoluci, and T. Mahmoodi. "Network slicing management & prioritization in 5G mobile systems". In: *European Wireless 2016; 22th European Wireless Conference*. May 2016, pp. 1–6.

[47] Ubuntu Juju. *Juju docs*. 2016.

[48] F. Kaltenberger et al. "OpenAirInterface: Open-source software radio solution for 5G". In: *European Conference on Networks and Communications (EU-CNC), Paris, France*. 2015.

[49] M. I. Kamel, L. B. Le, and A. Girard. "LTE Wireless Network Virtualization: Dynamic Slicing via Flexible Scheduling". In: *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*. Sept. 2014, pp. 1–5. DOI: `10.1109/VTCFall.2014.6966044`.

[50] M. I. Kamel, L. B. Le, and A. Girard. "LTE wireless network virtualization: Dynamic slicing via flexible scheduling". In: *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*. IEEE. 2014, pp. 1–5.

[51] K. Katsalis, N. Nikaein, and A. Huang. "JOX: An event-driven Orchestrator for 5G Network Slicing". In: *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. Apr. 2018, pp. 1–9. DOI: `10.1109/NOMS.2018.8406236`.

[52] K. Katsalis et al. "Network Slices toward 5G Communications: Slicing the LTE Network". In: *IEEE Communications Magazine* 55.8 (Aug. 2017), pp. 146–154. ISSN: 0163-6804. DOI: `10.1109/MCOM.2017.1600936`.

[53] K. Katsalis et al. "Virtual 802.11 wireless networks with guaranteed throughout sharing". In: *2015 IEEE Symposium on Computers and Communication (ISCC)*. IEEE. 2015, pp. 845–850.

[54] S. Kekki et al. *ETSI White Paper No. 28: MEC in 5G networks*. 2018.

[55] E. Keller et al. “Live migration of an entire network (and its hosts)”. In: *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*. ACM. 2012, pp. 109–114.

[56] S. Keranidis et al. “Enabling AGILE spectrum adaptation in commercial 802.11 WLAN deployments”. In: *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM. 2014, pp. 295–298.

[57] Y. Khadraoui, X. Lagrange, and A. Gravey. “Performance analysis of LTE-WiFi very tight coupling”. In: *13th IEEE Annual Consumer Communications Networking Conference (CCNC)*. 2016. DOI: 10.1109/CCNC.2016.7444757.

[58] R. Kokku et al. “CellSlice: Cellular wireless resource slicing for active RAN sharing”. In: *2013 Fifth International Conference on Communication Systems and Networks (COMSNETS)*. Jan. 2013, pp. 1–10. DOI: 10.1109/COMSNETS.2013.6465548.

[59] R. Kokku et al. “CellSlice: Cellular wireless resource slicing for active RAN sharing”. In: *2013 Fifth International Conference on Communication Systems and Networks (COMSNETS)*. IEEE. 2013, pp. 1–10.

[60] M. Kourtis et al. “T-NOVA: An Open-Source MANO Stack for NFV Infrastructures”. In: *IEEE Transactions on Network and Service Management* 14.3 (Sept. 2017), pp. 586–602. ISSN: 1932-4537. DOI: 10.1109/TNSM.2017.2733620.

[61] K. Kousias et al. “Building virtual 802.11 testbeds towards open 5G experimentation”. In: *Wireless Communications and Networking Conference (WCNC), 2016 IEEE*. IEEE. 2016, pp. 1–7.

[62] C. Y. Li et al. “Mobile Edge Computing Platform Deployment in 4G LTE Networks: A Middlebox Approach”. In: *{USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 18)*. 2018.

[63] *Linux Foundation, ONAP – Open Network Automation Platform*. Online], [http://onap.org/.

[64] D. Lopez-Perez et al. “Enhanced Intercell Interference coordination challenges in Heterogeneous Networks”. In: *IEEE Wireless Communications* 18.3 (June 2011), pp. 22–30.

[65] N. Makris et al. “Cloud-Based Convergence of Heterogeneous RANs in 5G Disaggregated Architectures”. In: *2018 IEEE International Conference on Communications (ICC)*. May 2018, pp. 1–6. DOI: 10.1109/ICC.2018.8422227.

[66] N. Makris et al. “Employing MEC in the Cloud-RAN: An Experimental Analysis”. In: *Proceedings of the 2018 on Technologies for the Wireless Edge Workshop*. New Delhi, India: ACM, 2018, pp. 15–19. ISBN: 978-1-4503-5931-3. DOI: 10.1145/3266276.3266281.

[67] N. Makris et al. “Enabling open access to LTE network components; the NITOS testbed paradigm”. In: *1st IEEE Conference on Network Softwarization (NetSoft)*. Apr. 2015. DOI: 10.1109/NETSOFT.2015.7116191.

[68] N. Makris et al. “Experimental Evaluation of Functional Splits for 5G Cloud-RANs”. In: *IEEE International Conference on Communications (ICC)*. 2017. DOI: 10.1109/ICC.2017.7996493.

[69] N. Makris et al. “Integrating NFV-MANO with Wireless Services: Experiences and Testbed Development”. In: *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. Nov. 2018, pp. 1–5. DOI: 10.1109/NFV-SDN.2018.8725785.

[70] N. Makris et al. “Measuring LTE and WiFi coexistence in Unlicensed spectrum”. In: *2017 European Conference on Networks and Communications (EuCNC)*. June 2017, pp. 1–6. DOI: 10.1109/EuCNC.2017.7980769.

[71] N. Makris et al. “Spectrum Coordination for Disaggregated Ultra Dense Heterogeneous 5G Networks”. In: *2019 European Conference on Networks and Communications (EuCNC)*. June 2019, pp. 512–517. DOI: 10.1109/EuCNC.2019.8801955.

[72] J. Malinen et al. “hostapd: IEEE 802.11 AP, IEEE 802.1 x”. In: *WPA/WPA2/EAP/RADIUS Authenticator* (2014).

[73] J. Martins et al. “ClickOS and the art of Network Function Virtualization”. In: *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*. 2014, pp. 459–473.

[74] A. Mayoral et al. “Multi-tenant 5G Network Slicing Architecture with Dynamic Deployment of Virtualized Tenant Management and Orchestration

(MANO) Instances". In: *ECOC 2016; 42nd European Conference on Optical Communication*. Sept. 2016, pp. 1–3.

[75] N. McKeown et al. "OpenFlow: enabling innovation in campus networks". In: *ACM SIGCOMM Computer Communication Review* 38.2 (2008), pp. 69–74.

[76] R. Mijumbi et al. "Management and orchestration challenges in network functions virtualization". In: *IEEE Communications Magazine* 54.1 (2016), pp. 98–105.

[77] D. L. Mills. "Internet Time Synchronization: the Network Time Protocol". In: *IEEE Transactions on communications* 39.10 (1991), pp. 1482–1493.

[78] *Multi-access Edge Computing*. [Online], https://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing.

[79] N. Nikaein. "Processing radio access network functions in the cloud: Critical issues and modeling". In: *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*. ACM. 2015.

[80] N. Nikaein et al. "Demo: Closer to Cloud-RAN: RAN as a Service". In: *21st Annual International Conference on Mobile Computing and Networking*. ACM. 2015.

[81] N. Nikaein et al. "Network Store: Exploring Slicing in Future 5G Networks". In: *Proceedings of the 10th International Workshop on Mobility in the Evolving Internet Architecture*. MobiArch '15. Paris, France: ACM, 2015, pp. 8–13. ISBN: 978-1-4503-3695-6. DOI: 10.1145/2795381.2795390. URL: http://doi.acm.org/10.1145/2795381.2795390.

[82] N. Nikaein et al. "OpenAirInterface 4G: an open LTE network in a PC". In: *20th Annual International Conference on Mobile Computing and Networking*. ACM. 2014.

[83] N. Nikaein et al. "OpenAirInterface: A Flexible Platform for 5G Research". In: *SIGCOMM Comput. Commun. Rev.* 44.5 (2014). DOI: 10.1145/2677046.2677053.

[84] *NITOS - Network Implementation Testbed using Open Source platforms*. [Online], https://nitlab.inf.uth.gr/NITlab/.

[85] *ONAP Support for 5G Wireless Network*. [Online], https://wiki.onap.org/pages/viewpage.action?pageId=15992173.

[86] L. Peterson et al. "Slice-based Federation Architecture (SFA)". In: *Working draft, version* 2 (2010).

[87] B. Pfaff et al. "The Design and Implementation of Open vSwitch". In: *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*. 2015, pp. 117–130.

[88] Protobuf-C. *"protobuf-c–C bindings for Google's Protocol Buffers–Google Project Hosting"*.

[89] T. Rakotoarivelo et al. "OMF: a control and management framework for networking testbeds". In: *ACM SIGOPS Operating Systems Review* 43.4 (2010), pp. 54–59.

[90] A. Raniwala, K. Gopalan, and T. Chiueh. "Centralized Channel Assignment and Routing Algorithms for Multi-channel Wireless Mesh Networks". In: *SIGMOBILE Mob. Comput. Commun. Rev.* 8.2 (Apr. 2004), pp. 50–65.

[91] A. Ravanshid et al. "Multi-connectivity functional architectures in 5G". In: *IEEE International Conference on Communications Workshops (ICC)*. May 2016. DOI: 10.1109/ICCW.2016.7503786.

[92] C. Richards. *Common Radio Resource Control for Cellular Radio and WiFi*. US Patent App. 13/707,184.

[93] M. Richart et al. "Resource allocation for network slicing in WiFi access points". In: *2017 13th International Conference on Network and Service Management (CNSM)*. Nov. 2017, pp. 1–4. DOI: 10.23919/CNSM.2017.8256046.

[94] M. Richart et al. "Resource Slicing in Virtual Wireless Networks: A Survey". In: *IEEE Transactions on Network and Service Management* 13.3 (Sept. 2016), pp. 462–476. ISSN: 1932-4537. DOI: 10.1109/TNSM.2016.2597295.

[95] C.B. Sankaran. "Network access security in next-generation 3GPP systems: A tutorial". In: *IEEE Communications Magazine* 47.2 (2009), pp. 84–91.

[96] R. Santos and A. Kassler. "A SDN controller architecture for Small Cell Wireless Backhaul using a LTE Control Channel". In: *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. June 2016, pp. 1–3. DOI: 10.1109/WoWMoM.2016.7523544.

[97] O. Sefraoui, M. Aissaoui, and M. Eleuldj. "OpenStack: toward an open-source solution for cloud computing". In: *International Journal of Computer Applications* 55.3 (2012), pp. 38–42.

[98] R. Sherwood et al. "Flowvisor: A network Virtualization Layer". In: *OpenFlow Switch Consortium, Tech. Rep* 1 (2009), p. 132.

[99] *SiRRAN Communications NetCORE technology*. [Online], https://www.sirran.com/technology/.

[100] S. Soltesz et al. "Container-based Operating System Virtualization: a scalable, high-performance alternative to Hypervisors". In: *ACM SIGOPS Operating Systems Review*. Vol. 41. 3. ACM. 2007, pp. 275–287.

[101] D. Stavropoulos et al. "Design, Architecture and Implementation of a Resource Discovery, Reservation and Provisioning Framework for Testbeds". In: *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE. 2015, pp. 48–53.

[102] T. Stockhammer et al. "MPEG systems technologies - part 6: Dynamic adaptive streaming over HTTP (DASH)". In: ISO/IEC, MPEG Draft International Standard. 2011.

[103] K. Tan et al. "Sora: high-performance software radio using general-purpose multi-core processors". In: *Communications of the ACM* 54.1 (2011), pp. 99–107.

[104] Blu Wireless Technology. *Blu Wireless Technology WiGig products*. [Online], http://www.bluwirelesstechnology.com/.

[105] *The Fed4FIRE project*. [Online], https://fed4fire.eu.

[106] R. Vilalta et al. "Hierarchical SDN orchestration of wireless and optical networks with E2E provisioning and recovery for future 5G networks". In: *2016 Optical Fiber Communications Conference and Exhibition (OFC)*. Mar. 2016, pp. 1–3.

[107] A. Virdis et al. "Performance analysis of OpenAirInterface system emulation". In: *3rd International Conference on Future Internet of Things and Cloud (FiCloud), 2015*. IEEE, 2015.

[108] J. Wang et al. "A demonstration of a new TCP Congestion Control Algorithm over LTE and other challenging networks". In: *16th Annual International Conference on Mobile Computing and Networking*. ACM. 2010.

[109] D. Wubben et al. "Benefits and impact of cloud computing on 5G signal processing: Flexible centralization through cloud-RAN". In: *IEEE signal processing magazine* 31.6 (2014).

[110] Y. Yiakoumis et al. "Slicing Home Networks". In: *Proceedings of the 2Nd ACM SIGCOMM Workshop on Home Networks*. HomeNets '11. Toronto, Canada: ACM, 2011, pp. 1–6. ISBN: 978-1-4503-0798-7. DOI: 10.1145/2018567.2018569. URL: http://doi.acm.org/10.1145/2018567.2018569.

[111] G. Yilma et al. *On the Challenges and KPIs for Benchmarking Open-Source NFV MANO Systems: OSM vs ONAP*. 2019. arXiv: 1904.10697 [cs.NI].

[112] F. Z. Yousaf et al. "NFV and SDN-Key Technology Enablers for 5G Networks". In: *IEEE Journal on Selected Areas in Communications* 35.11 (2017), pp. 2468–2478.

[113] K. Zhang et al. "Energy-efficient offloading for Mobile Edge Computing in 5G Heterogeneous Networks". In: *IEEE access* 4 (2016), pp. 5896–5907.

[114] Y. Zhiling et al. "White Paper of Next Generation Fronthaul Interface v1.0". In: *China Mobile Research Institute, Tech. Rep* (2015).